



TRABAJO FIN DE GRADO

Grado en Ingeniería Electrónica Industrial y Automática

Proteus Plóter Multiherramienta Imprimible Open Source

Autor : Francisco Javier del Álamo García

Tutor : Andrea Bellucci

Fecha de Presentación: Julio 2015

Índice general

Agradecimientos	4
Resumen	5
Abstract	6
1. Introducción	7
1.1. Motivación	9
1.2. En el documento	9
2. Análisis del estado del arte, sobre fabricación digital y plóteres	10
2.1. Fabricación digital	10
2.1.1. Que entendemos por fabricación digital	10
2.1.2. Los beneficios de la fabricación digital	11
2.1.3. Máquinas de fabricación digital	11
2.1.3.1. Industriales	11
2.1.3.2. Domésticas	12
2.1.4. Software de fabricación digital	12
2.1.5. El boom de las impresoras 3D	13
2.1.6. Grabadores láser	13
2.2. Comparativa de plóteres	14
3. Análisis del problema: requisitos y restricciones	16
3.1. Normativas legales a cumplir	16
3.2. Soluciones propuestas - Alternativas de diseño	17
3.2.1. Plóter portátil basado en el polargraph	17
3.2.2. Plóter fijo con estructura de pantógrafo	18
3.2.3. Plóter portable con estructura de paralex	19
3.2.4. Plóter fijo X-Y	20
3.2.5. Plóter X-Y portable	20
3.3. Elecciones de hardware, software y modo de fabricación	21
3.3.1. Elección del controlador	21
3.3.2. Elección de la integración de las herramientas	22
3.3.3. Elemento tractor: Correas o varillas roscadas	22
3.3.4. Aplicaciones para smartphones, tablets y ordenadores	23
3.3.5. Interfaz	24
3.3.6. Elección del modo de fabricación	24

4. Descripción del Hardware	25
4.1. Unidad de Control	25
4.1.1. Arduino Uno	26
4.1.2. CNC Shield	27
4.1.2.1. Drivers pololu	28
4.2. Motores NEMA 17	29
4.3. Fuente de alimentación	30
4.4. Miniordenador con el servidor de la webapp: Raspberry Pi	31
4.4.1. USB Wifi	32
4.5. Transformador	32
5. Proteus	33
5.1. Piezas Impresas	33
5.1.1. Diseño: OpenSCAD	33
5.1.2. Descripción de las piezas	34
5.1.3. Fallos de impresión y mejoras aplicadas	38
5.2. Piezas estándar de ferretería	39
5.3. Esquema de montaje de la electrónica	40
6. Módulo de herramienta: Surya	42
6.1. Hardware específico	42
6.1.1. Láser	42
6.1.2. Driver	43
6.2. Pieza de agarre al carro de la deslizadera	45
6.3. Seguridad	45
6.4. Posibles usos	46
6.4.1. Creación de PCBS	46
6.4.2. Personalización de objetos	47
6.4.2.1. Personalización de ropa	47
6.4.3. Diseño y grabado de tableros de juego en madera	48
6.4.4. Creación de lámparas	48
7. Módulo de herramienta: Ptah	49
7.1. Hardware específico	49
7.1.1. Servomotor	49
7.1.2. Driver	50
7.2. Pieza de agarre al carro de la deslizadera	52
7.3. Seguridad	52
7.4. Posibles usos	53
8. Planificación	55
8.1. Planificación inicial	55
8.2. Planificación real	57
9. Presupuesto	58
9.1. Costes de desarrollo	58
9.1.1. Coste de material	58
9.1.2. Coste de herramientas	60

9.1.3. Coste personal	61
9.1.4. Coste total de desarrollo del proyecto	62
10. Conclusiones	63
10.1. Conclusión personal	64
10.2. Estado actual	65
10.2.1. Requisitos planteados frente a requisitos cumplidos	65
10.3. Mejoras futuras	65
10.3.1. Crear un diseño portable para grandes medidas	65
10.3.2. Diseñar los drivers de las herramientas como PCBS listas para soldar	66
10.3.3. Crear una red inalámbrica local que lleve directamente a la aplicación web	66
Bibliografía	66
A. Anexo1: Servidor de GRBLWeb en la Raspberry Pi con login	69
B. Anexo2: Piezas impresas diseñadas en Open SCAD	75
B.1. Piezas de Proteus	75
B.1.1. Cojinetes para los rodamientos: Rod.scad	75
B.1.2. Soporte superior de los motores: top.scad	76
B.1.3. Soporte inferior del motor derecho: bottomright.scad	77
B.1.4. Soporte inferior del motor izquierdo: bottomleft.scad	78
B.1.5. Unión motor-varilla superior: uniontop.scad	79
B.1.6. Unión motor-varilla inferior: unionbottom.scad	80
B.1.7. Soporte izquierdo de la deslizadera: leftsupport.scad	81
B.1.8. Soporte derecho de la deslizadera: rightsupport.scad	82
B.1.9. Deslizadera de la herramienta: toolsupport.stl	83
B.1.10. Pieza de sujección del láser: laserholder.stl	84
B.1.11. Pieza de sujeción del servo: servoholder.stl	85
C. Anexo3: Diagrama de la electrónica	86
D. Anexo4: Diagramas de Gantt	88

Agradecimientos

Me gustaría aquí dar gracias a todas aquellas personas que, de una manera u otra, me han ayudado durante mi carrera. No solo en la realización de este proyecto, sino a llegar hasta aquí.

En primer lugar quiero dar las gracias mi padre, Francisco Javier, y a mi madre, Gloria, por ser guías y apoyos constantes en mi vida y por enseñarme tantas cosas, muchas veces con su ejemplo y actitud, sobre como afrontar día a día los retos que la vida nos plantea. Ojalá sea capaz de no olvidar nunca sus enseñanzas y consejos, aplicarlos y transmitirlos.

Quiero dar las gracias a mis hermanas, Elisa y Teresa, por conseguir convertir algunos días mediocres en días memorables. A mis abuelos, por cuidarme siempre y enseñarme tanto, y a toda mi familia, por demostrarme siempre un cariño sincero e incondicional que nunca podré agradecer lo suficiente.

Además me gustaría agradecer a mi novia, Susana, toda su ayuda con esta memoria y proyecto y, por supuesto, su apoyo y comprensión en todo momento. A mis amigos, compañeros de estudios y de trabajo, quiero darles las gracias por hacer inolvidable esta etapa de mi vida en la universidad.

Y, cómo no, quiero dar las gracias a Juan y Alberto por darme la oportunidad de aprender de ellos e iniciarme en el campo de las impresoras 3D, y por crear y desarrollar la comunidad Clone Wars que tanto me aporta a mi y a muchos más.

Finalmente, quiero dar las gracias a todos los docentes que me han llevado a ser quien soy, y en especial a mi tutor, Andrea, por su dedicación, trabajo y buen consejo durante el desarrollo de este proyecto.

Gracias, sin vosotros esto no sería posible.

Resumen

En los últimos años se ha producido un auge en la difusión de herramientas de fabricación digital doméstica, pero aún no existe un plóter open source de fabricación personal y bajo coste, lo que beneficiaría enormemente a la comunidad maker, permitiendo la creación de PCBS, maquetas, prototipos y, por que no, obras de arte. Por eso he desarrollado Proteus, aspirando a crear un producto completo tal y como yo lo querría si fuese a comprarlo.

Proteus es un sistema de plóter capaz de trazar trayectorias controlado de forma remota a través de un entorno web. Esto le permite convertirse fácilmente, en función de la herramienta y posibles extensiones en una máquina de grabado láser, corte láser, corte de vinilo, dibujo con bolígrafo, impresora 3D, escáner 3D...

Ha sido diseñado para ofrecerse como producto en kit para uso doméstico, desarrollo e investigación.

Proteus es de sencillo ensamblaje y para utilizar la aplicación que sirve para controlarlo solo es necesario tener un dispositivo con un navegador que soporte HTML5.

La aplicación para el control remoto funciona independientemente del dispositivo y del sistema operativo. Se puede utilizar en ordenadores, tablets y smartphones sin necesidad de agregar en ellos ningún software.

Es importante señalar que Proteus es un proyecto abierto, en el que el usuario tiene acceso a todo el código fuente. Para facilitar el desarrollo de nuevas herramientas se han creado y documentado dos herramientas, el módulo láser Surya y el módulo de dibujo Ptah.

Actualmente hay un prototipo funcional del plóter y de dos herramientas, Surya y Ptah. También es plenamente funcional el control del plóter mediante una webapp sobre un servidor de node.js en la Raspberry Pi.

Abstract

In the last years there has been a boom in the diffusion of home digital fabrication tools, but there's not yet an opensource plotter for home fabrication at a low cost, and this would benefit the makers community, allowing PCBS creation, scale models, prototypes and, why not, art pieces. So I developed Proteus, aspiring to create a complete product, as I will like it as a customer.

Proteus is a plotter system able to make paths remotely controlled by a webapp. This let it easily work, depending on the tool and possible extensions, as a laser engraver, laser cutter, vinyl cutter, pen grapher, 3D printer, 3D scanner...

It has been designed to be offered as a kit for personal use, development and investigation.

Proteus is easy to assemble, and in order to use the webapp to control it you only need a device that accept HTML5.

The application for the remote control work independently of the device and the operative system. It can be used with desktop pc's, laptops, tablets and smartphones without needing to install any software.

Is important to notice that Proteus is an open project, where the user has access to all the source code. To help in the development of new tools, some has been developed and documented.

Nowadays there is a functional prototype of the plotter Proteus and some tools, specifically the laser engraver (Surya) and the pen grapher (Ptah). It's also fully functional the plotter control through a webapp in a node.js server on the Raspberry Pi.

Capítulo 1

Introducción

Comunidad Maker

La comunidad maker, o simplemente seguidores del movimiento maker, son todas aquellas personas que se sienten identificados con una serie de principios y valores (Haz. Comparte. Aprende. Apoya.). Gracias a la filosofía de conocimiento libre compartida por la comunidad, que podríamos reducir a compartir el conocimiento y aprender constantemente, las ideas evolucionan incesantemente adaptándose a las necesidades de cada maker. En esta comunidad se valora mucho el aprender a través de la experiencia mediante proyectos DIY (Do It Yourself, Hazlo tu mismo) por lo que cada individuo tiene a su disposición diferentes herramientas que le permiten crear sus propios productos. Actualmente es una de las tendencias más importantes en el desarrollo doméstico de proyectos. Al compartir y desarrollar las ideas con la comunidad, todo el mundo gana, pues otras personas colaboran con otros puntos de vista y conocimientos transversales.

Introducción a la fabricación digital open source

La fabricación digital incluye tanto el software de diseño asistido con el objetivo de automatizar los procesos de fabricación en las máquinas correspondientes como las propias máquinas. Desde impresoras 3D, tornos, fresadoras, máquinas de corte láser, máquinas de roto-moldeo, y casi cualquier máquina que podamos imaginar asociada a un proceso de fabricación, todas pueden ser controladas para planificar el proceso de producción, determinando el tiempo y recursos que se necesitarán para acometer la tarea, así como una estimación de los costes.

En el ámbito doméstico despunta la fabricación digital open source, que si bien no consta todavía de todas las máquinas y software de los que dispone la industria, va ganando fuerza, pues los costes son mucho menores. Ya ha sido denominada como "La revolución industrial del siglo XXI", y las posibilidades que ofrecen estas máquinas y software promovidos por la comunidad maker, crecen día a día.

Arduino

El entorno de desarrollo abierto Arduino es uno de los más utilizados por la comunidad maker en la construcción de los muchos y variados proyectos que desarrollan, ya que es un entorno muy flexible y de muy bajo coste. Además, la familia de dispositivos compatible con este entorno es muy variada, lo que permite que se adapte a las necesidades de cada usuario y su programación es suficientemente sencilla como para aprender a partir de ejemplos.

Proteus: Breve descripción

Proteus es un sistema de plóter controlado de forma remota a través de un entorno web. Está pensado como una herramienta para la comunidad maker, buscando que sea accesible y fácil de montar. Al ser multiherramienta, sus aplicaciones dependerán de los múltiples cabezales de herramientas que se desarrollen y de la creatividad de la comunidad.

Está diseñado para adaptarse a las necesidades de cada usuario, pues según el tipo de proyectos que se quieran realizar, cada maker tendrá unas necesidades concretas. Así está diseñado de manera que las dimensiones de Proteus dependen únicamente de las varillas metálicas que componen su estructura, por lo que es muy fácil reducirlas o ampliarlas. Por supuesto, una ampliación grande podría requerir motores más potentes.

Es importante recordar que está concebido como un kit, por lo que requiere de montaje, conexionado, configuración, calibrado... Todas estas tareas, generalmente inusuales en un producto, son aceptadas como completamente normales en la comunidad maker, a la que va dirigido.

Proteus es un proyecto pensado para aportar valor a la comunidad siendo una nueva herramienta sencilla y económica, pero igualmente se nutre de ella. De esta forma, la aplicación web es un entorno open source llamado GRBLWeb, de xyzbots. Se ha instalado como servidor en una Raspberry Pi y por seguridad se ha generado un servidor que redirige a una pantalla de login automáticamente y que da paso al control de Proteus mediante GRBLWeb. De esta forma, controlar remotamente el dispositivo solo requiere un navegador web que soporte HTML5 y se puede utilizar en pc, tablet o smartphone con cualquier sistema operativo como Linux, MacOSX, Windows, Android, iOS, WindowsPhone...

Orientado como está a una comunidad de gente innovadora, el código fuente de Proteus es abierto, de manera que cualquiera tiene acceso a el, y el desarrollo de cabezales de herramientas está documentado para permitir que la comunidad pueda crear nuevas herramientas con la mayor facilidad posible.

Antes de comenzar con el desarrollo de Proteus se hizo un estudio de mercado para evaluar las características de dispositivos similares, de estos se extrajeron las siguientes conclusiones: son productos centrados en una funcionalidad, es decir, no son multiherramienta, los precios oscilan entre 600-6500 €, son complejos y de alta potencia y por todo ello son difíciles de entender como máquina de fabricación personal.

1.1. Motivación

El proyecto Proteus nació a principios de 2014 con el objetivo de crear una máquina de corte láser muy económica y que cualquiera pudiese montar en su casa. No mucho tiempo después, se vio que sería mucho más interesante si se desarrollaba un sistema de herramientas intercambiables. De este modo, la idea original está representada por el módulo láser, pero sus funcionalidades pueden cambiar completamente al cambiar de herramienta. Con la idea de desarrollar un producto que cumpliera mis expectativas como consumidor, se implementó el control remoto multiplataforma, se minimizó el número de piezas necesarias para su construcción, se simplificó en la medida de lo posible la electrónica a utilizar, y en definitiva, se procuró que el trabajo a realizar una vez obtenido el kit fuese sencillo y mínimo.

Por ahora este proyecto se encuentra en su primera versión, pero se prevén cambios y desarrollos que se explicarán en el apartado de futuras mejoras

1.2. En el documento

En el próximo capítulo se presentará el estado del arte y a continuación, en el capítulo 3, requisitos, restricciones y soluciones alternativas que se han planteado para resolver el diseño de Proteus.

A continuación, en el capítulo 4 se procederá a describir el hardware.

En los capítulos 5, 6 y 7 se describirán respectivamente el plóter Proteus y las herramientas láser y rotulador (Denominadas Surya y Ptah).

Posteriormente en el capítulo 8 veremos la planificación propuesta para el desarrollo de este proyecto, así como su ejecución y correcciones para llevarla a cabo.

En el capítulo 9 se encuentra el presupuesto para el prototipo y un presupuesto estimado para su desarrollo.

En el capítulo 10 se detallan las conclusiones obtenidas tras el desarrollo del proyecto, así como las mejoras y futuras implementaciones que se proponen como línea de trabajo.

Por último, en el capítulo 11, se encuentra la bibliografía, seguida de los anexos.

El Anexo 1 trata de las modificaciones de software necesarias para hacer funcionar el servidor de login.

El Anexo 2 contiene el código de todas las piezas diseñadas en OpenSCAD para Proteus.

El Anexo 3 contiene, para una mayor claridad expositiva, una ampliación del diagrama de la electrónica del plóter del capítulo 5.

El Anexo 4 contiene ampliaciones apaisadas de los diagramas de Gantt de la planificación para que se puedan visualizar sin problema.

Capítulo 2

Análisis del estado del arte, sobre fabricación digital y plóteres

2.1. Fabricación digital

El origen de la tecnología de fabricación digital se encuentra a mediados del siglo XX, al converger las máquinas de control numérico con las tecnologías de computación. Al abaratare en los 80 los ordenadores en las industrias del transporte se consolidaron las prácticas de CAD/CAE/CAM.

Actualmente la maquinaria CNC se usa de forma habitual en la industria. Además proliferan en los centros de estudios especializados programas formativos relacionados con estas tecnologías y de igual forma se construyen laboratorios de fabricación digital por todo el mundo.

2.1.1. Que entendemos por fabricación digital

Entendemos por fabricación digital el uso de un sistema basado en un ordenador que comprende todos los programas que se utilizan en la industria para la simulación, visualización 3D, análisis y herramientas de colaboración para definir un producto y todos los procesos de fabricación conjuntamente, incluyendo la planificación integrada aplicación, control y mejora continua de los mismos.

Todo esto permite a las empresas que fabrican lograr cumplir con sus objetivos de volumen, con los tiempos de salida al mercado previstos para su producto e incluso ahorrar costes al no tener que realizar cambios posteriores.

Estos sistemas digitales de producción permiten a los ingenieros definir en entornos virtuales sus procesos de fabricación de forma muy detallada, incluyendo los recursos y las herramientas, e incluso líneas de ensamblaje y centros de trabajo, todo dispuesto sobre un diseño virtual de sus instalaciones.

2.1.2. Los beneficios de la fabricación digital

La tecnología de fabricación digital puede ayudar a las empresas a mejorar la planificación de la fabricación y los procesos de producción. Permite asociar productos, instalaciones, procesos y recursos, de manera que puedan llevarse a cabo procesos productivos con un enfoque global acorde con el diseño de producción.

Además ayuda a reducir los costes de puesta en marcha validando de forma virtual la robótica y automatización de los programas asociados al ciclo productivo, asegurando antes de la producción que se trabajara sobre un diseño óptimo de flujo de materiales y rendimiento. Se puede utilizar la fabricación digital para apoyar sistemas de control de calidad como Six-Sigma

Al utilizarse programas de inspección verificables se facilita el intercambio de datos de calidad en toda la fábrica y se tiene acceso en tiempo real a todos los datos del ciclo de producción.

2.1.3. Máquinas de fabricación digital

2.1.3.1. Industriales

En la industria podemos encontrar múltiples máquinas de fabricación digital, a destacar:

Tornos CNC: Un torno es una máquina que permite mecanizar piezas por revolución. Mientras la pieza a mecanizar gira en un cabezal, se desplazan una o varias herramientas que dan forma a la pieza por arranque de viruta.

Fresadoras CNC: Son máquinas que trabajan por arranque de viruta mediante una herramienta rotativa de varios filos (la denominada fresa). Permiten trabajar sobre superficies planas y curvas, y sobre diversos materiales como madera y acero.

Impresoras 3D: Estas máquinas funcionan mediante tecnología aditiva. Existen varias tecnologías que hacen posible la impresión 3D si bien las más empleadas son las de FDM (Deposición de material fundido), SLS (Sintetizado selectivo por láser) y SLM (Fundido selectivo por láser). Estas máquinas trabajan con plásticos y resinas.

Máquina de corte o grabado láser: Son máquinas que funcionan posicionándose en el plano mediante diversos mecanismos y accionando un láser. Generalmente las máquinas industriales usan un láser de CO_2 que son muy potentes pero hace que se necesiten sistemas de ventilación. Pueden cortar y grabar madera, metales, plásticos. . .

Máquina de deposición CNC: Estas máquinas succionan y depositan componentes electrónicos SMD sobre PCBS. Son muy utilizadas y de alta precisión.

Máquina de corte por chorro de agua: Las máquinas de corte por chorro de agua funcionan por impacto del chorro de agua a gran presión y velocidad

portando un material abrasivo que producen una desgaste del material hasta generar el corte. Puede cortar prácticamente cualquier material.

2.1.3.2. Domésticas

En la fabricación digital doméstica despuntan las impresoras 3d, pero también existen cortadoras láser y maquinas cnc. Estas máquinas son de menor potencia y precisión que las maquinas industriales, pero también implican un coste muy inferior, de entre el 1 % y el 5 %.

Estas máquinas son open source, y su bajo coste viene dado fundamentalmente por la elección de piezas estándar, componentes más económicos y de menor potencia y el hecho de que en la mayoría de los casos vienen en forma de kit, por lo que el comprador debe estar dispuesto a construir la estructura de la máquina y muchas veces no solo montar y programar la electrónica, sino incluso soldar todos los componentes electrónicos a la PCB.

De esta forma, el comprador está obligado a adquirir una serie de habilidades y conocimientos técnicos, y el vendedor puede ahorrar enormemente en costes, pues su función fundamental es el diseño de la máquina y su electrónica y la compra al por mayor de los componentes para distribuirlos en forma de kit. En algunos casos en los que no todas las piezas empleada son estándar, el vendedor puede externalizar la fabricación de esas piezas o encargarse de la misma, siendo esta ultima opción más frecuentemente elegida cuando esas piezas se pueden imprimir en impresoras 3D de bajo coste.

La principal ventaja de disponer de máquinas de fabricación digital doméstica es que cualquiera puede fabricar sus propios diseños adaptados a sus necesidades, sin desplazarse de su casa y sin costes de intermediarios.

Actualmente existen múltiples diseños de impresoras 3D, cortadoras/grabadoras láser y fresadoras CNC.

2.1.4. Software de fabricación digital

Este software es el empleado tanto para el diseño de las piezas (Que comentaremos a continuación) como para el control y la automatización de procesos (Que muchas veces son propietarios de las empresas que venden todo el sistema integrado).

Diferentes máquinas pueden requerir de programas con características muy diferentes, pero vamos a distinguir entre los programas para las fresadoras CNC y las impresoras 3D que necesitan de un software capaz de generar una representación tridimensional de la pieza que estemos diseñando y los programas para para las máquinas de corte/grabado láser que son más sencillos ya que muchos programas de control permiten como entrada una imagen en escala de grises, aunque el diseño más recomendado es el vectorial.

Dado que el plóter Proteus está diseñado junto con una herramienta para hacer grabado/corte láser y otra para realizar imágenes, el software que se utilizará

para realizar los distintos diseños será software de diseño vectorial. Existen dos alternativas, una comercial y otra open source, muy utilizadas:

Adobe Illustrator es un programa de gráficos vectoriales pensado para la creación artística de dibujo para impresión, vídeo, publicación en la Web y dispositivos móviles. Desarrollado y comercializado por Adobe Systems, este programa es apropiado para diseñar trayectorias que se usen con máquinas de corte o grabado.

Inkscape es un programa de gráficos vectoriales en formato SVG, gratuito, libre y multiplataforma. Su objetivo fundamental es permitir a los usuarios elaborar gráficos en formato vectorial escalable. Soporta la creación de trayectorias que se pueden utilizar en máquinas de fabricación digital de corte y grabado.

2.1.5. El boom de las impresoras 3D

El proyecto RepRap nace en 2005, fundado por Adrian Bowyer, ingeniero mecánico de la Universidad de Bath. El 9 de Febrero de 2008 se completó la primera RepRap 1.0 Darwin funcional. Tras esto el proyecto RepRap creció de forma incesante. En septiembre del mismo año ya había más de 100 impresoras de este modelo activas. Poco más de un año después se rediseñó y nació la Prusa 2 "Mendel".

Durante los últimos años se ha producido una enorme proliferación de las impresoras 3D domésticas como medio de fabricación digital doméstico. Si bien parte de esta proliferación es debida a que existía una comunidad maker que demandaba soluciones para la creación de prototipos, uno de los mayores hitos que han disparado este boom ha sido el vencimiento de la patente de la tecnología de deposición de material fundente, y por otra parte, la aparición de una serie de diseños de impresoras 3D libres.

Entre estas hay que destacar la RepRap Mendel (Ed Sells - Octubre/2009), que pasó de una forma cúbica a un prisma triangular haciéndola más estable, la Printbot (Brook Drumm - Septiembre/2012) que requiere menos piezas impresas y baja los motores del eje z a la base y la Prusa i3 (Josef Prusa - Septiembre/2012) que añade un marco que se sujeta a la base de la impresora y sirve para soportar las varillas verticales del eje z.

2.1.6. Grabadores láser

Si bien existen algunos proyectos de grabadoras láser, aún no se ha generado ese boom que si han experimentado las impresoras 3D, tal vez por su alto coste y grandes dimensiones. Si es cierto, sin embargo, que en el tiempo que ha tomado el desarrollo de este proyecto se ha observado que se han comenzado proyectos relacionados. Tal vez 2015 sea el año del láser.

2.2. Comparativa de plóteres

Actualmente no existen muchos productos similares a Proteus, pero si muchos productos que tienen algunas cosas en común, por lo que se reseñan los más significativos por precios, dimensiones y herramientas, que son el proyecto Lasersaur [1], Piccolo [2], Shapeoko [3], Mr.Beam [4] y Cyclone [5].

Nombre	Herramienta	Dimensiones	Area de trabajo	Proporción
Lasersaur	CO2 Láser	1700x1080mm	1220x610 mm	40.5 %
Piccolo	Bolígrafo	120x60mm	50x50mm	35 %
Shapeoko	Fresa	725x600mm	425x425mm	20 %
Mr. Beam	Diodo Láser	590x440mm	297x216mm	25 %
Cyclone PCB-Factory	Taladro	250x300mm	100x160mm	22 %
Proteus Plotter	Diodo Láser	800x600mm	500x500mm	52 %

Tabla 2.1: Máquinas similares a Proteus A

Nombre	Materiales de trabajo
Lasersaur	Papel, cartón, madera, espuma, metacrilato, fieltro...
Piccolo	Papel, cartulina...
Shapeoko	Madera, plásticos, piedra. cobre, bronce, acero, hierro...
Mr. Beam	Corcho, nylon, seda, algodón, poliéster...
Cyclone PCB-Factory	PCB
Proteus Plotter	Grabar: Madera, metacrilato, corcho... Cortar: Papel, cartulina, poliéster, seda...

Tabla 2.2: Máquinas similares a Proteus B

Nombre	Potencia	Fecha	Precio (aprox)
Lasersaur	100W	9-7-2010	6408 €
Piccolo	-	9-12-2014	94 €
Shapeoko	840W	5-2015 (Prevista)	1246 €
Mr.Beam Jr.	200mW	7-6-2014	570 €
Cyclone PCB-Factory	300W	5-2013	237 €
Proteus Plotter	250mW	-	200 €, idealmente 150 €

Tabla 2.3: Máquinas similares a Proteus C

Nombre	Web
Lasersaur	www.lasersaur.com
Piccolo	https://github.com/DiatomStudio/Piccolo/wiki/Piccolo-Bill-of-Materials
Shapeoko	http://www.shapeoko.com/
Mr.Beam Jr.	http://www.mr-beam.org/
Cyclone PCB-Factory	https://github.com/carlosgs/Cyclone-PCB-Factory
Proteus Plotter	-

Tabla 2.4: Máquinas similares a Proteus D

En los datos presentados se puede observar que los diodos láser tienen consumos muy inferiores al láser de CO_2 y que en cualquier caso, las herramientas mecánicas, que funcionan con contacto físico desbastando material, requieren de una potencia mucho mayor.

También se puede ver que, en general, las dimensiones del área de trabajo no son muy grandes, incluso cuando las dimensiones de la máquina son considerables. De hecho, de media, el área de trabajo es tan solo el 28.5 % del área de las máquinas.

Si nos fijamos en las fechas en las que estas máquinas fueron anunciadas, podemos ver en los últimos años un incremento de los proyectos de plóteres como máquina de fabricación doméstica. Esto lleva a pensar que en un futuro cercano este tipo de máquinas podría, como ya hicieron las impresoras 3D, popularizarse en los entornos de los laboratorios de fabricación.

En cuanto al factor precio, ni la máquina Lasersaur ni la máquina Shapeoko tienen precios acordes a lo que se esperaría de una máquina de fabricación digital doméstica. El resto de máquinas no tienen precios excesivos, pero tampoco son muy económicas.

Tras estudiar estos datos podemos ver que las únicas máquinas que podríamos considerar de bajo coste son la máquina Piccolo, que tiene un área de trabajo muy reducida, y la máquina Cyclone PCB-Factory, que tiene unas dimensiones apropiadas para crear PCBS pero se quedaría corta para corte/grabado láser.

La máquina Mr.Beam Jr. tiene un área de trabajo aproximadamente igual al área de un DIN-A4, y aunque solo tiene una herramienta, sería posible diseñar otras herramientas compatibles, por lo que su mayor inconveniente es el precio.

Basándonos en lo anterior, se establecerán más adelante una serie de requisitos para el desarrollo de Proteus, pero para mostrarlos de forma comparativa con las máquinas anteriores se han incluido en las tablas algunos de los posibles requisitos que este estudio nos sugiere.

Capítulo 3

Análisis del problema: requisitos y restricciones

Una vez analizados los distintos productos con cosas en común a Proteus, se han decidido los requisitos y restricciones del proyecto en base a los objetivos planteados en primera instancia.

El objetivo principal de este proyecto era desarrollar un plóter multiherramienta open source que se pudiese construirse, utilizarse y transportarse con comodidad. Concretamente se quería que las piezas fuesen mínimas, que el cambio de herramienta fuese lo más sencillo posible y que se pudiese utilizar remotamente desde cualquier dispositivo sin necesidad de agregar software.

Por supuesto se requiere que el coste sea muy bajo para hacerlo lo más accesible posible a la comunidad.

Se requiere que la mecánica del movimiento sea lo más precisa posible sin ser muy compleja, pues cuanto más sencilla sea, menos posibilidades de fallo habrá. Se requiere que las dimensiones sean flexibles para que cualquiera pueda adaptarlo a sus necesidades, que todo el software utilizado sea abierto (para permitir que cualquiera pueda estudiar los fuentes) y que su montaje se realice con herramientas estándar.

Además se busca que montar la electrónica y cambiar las herramientas, así como cargar el software, no requiera ningún conocimiento específico y que con solo seguir la guía de montaje se llegue a obtener el producto final a partir del kit.

Por seguridad, se requiere que el control de la máquina se realice identificando previamente al usuario.

3.1. Normativas legales a cumplir

La electrónica básica de Proteus, (El Arduino UNO, la placa CNC Shield y los drivers pololu, además de la fuente de alimentación y la Raspberry Pi junto el USB Wifi) son todo componentes comerciales que cumplen todas las normativas de la Unión Europea de protección eléctrica y de ondas.

Las piezas y en general todo Proteus, al ser un kit para makers y no un producto cerrado, debe avisar de los peligros y normas de seguridad pertinentes, pero deja al usuario el como cumplirlas. Por supuesto, es muy importante señalar que siempre que se trabaje con láseres se deberá utilizar la protección ocular correspondiente. Del mismo modo, se debe indicar que ni la mecánica ni la electrónica deberán ser manipuladas sin apagar la fuente de alimentación y desenchufarla de la corriente eléctrica.

La electrónica diseñada específicamente para el proyecto (Los drivers de Surya y Ptah) deberán someterse a pruebas para poder comercializarla como producto terminado bajo la normativa europea. No obstante, como son un prototipo, se incluirán en el kit los componentes comerciales necesarios para su construcción junto con la documentación pertinente, advirtiendo en todo caso de que no es un producto final.

3.2. Soluciones propuestas - Alternativas de diseño

Una vez analizados los requisitos, comenzamos la fase de diseño. Aquí veremos las distintas alternativas planteadas para realizar el plóter y los mayores inconvenientes de las soluciones no elegidas.

3.2.1. Plóter portátil basado en el polargraph

Como se puede apreciar en la figura esta propuesta basa su movimiento en la máquina conocida como polargraph, que es una máquina de dibujo colgante. Utiliza tan solo dos motores, es portátil y se puede adaptar a múltiples dimensiones. Problemas de esta solución son una clara falta de precisión (debida a que habitualmente se realiza colgando de hilos de nylon), una gran dificultad para auto calibrar la máquina en función de las proporciones de la superficie donde se sitúe y fallos de seguridad al usar herramientas como láseres o taladros colgados en vertical.

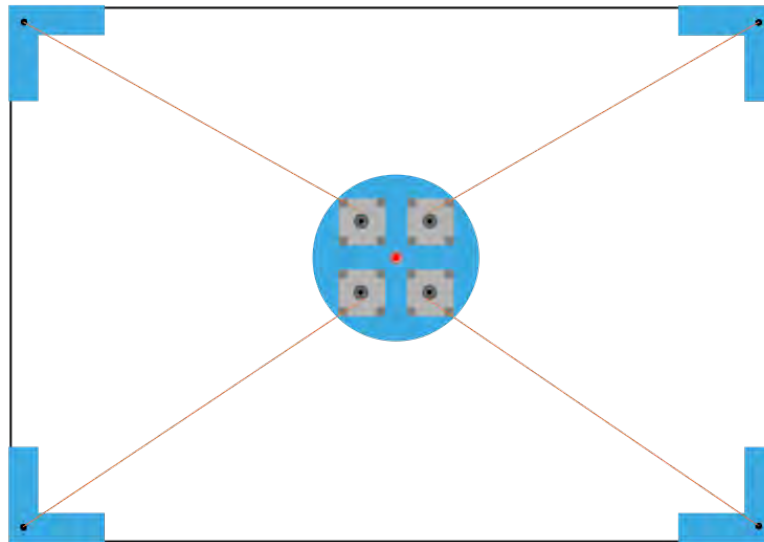


Figura 3.1: Ploter polargraph

3.2.2. Plóter fijo con estructura de pantógrafo

Utilizando la misma estructura de la que dispone un pantógrafo, se puede realizar un plóter que posicione una herramienta en el plano. Tiene la ventaja de necesitar solo dos motores, muy pocos rodamientos y tornillos y muy pocos elementos más. Entre los inconvenientes nos encontramos que el control se realiza en base al ángulo de giro del motor, lo que nos obligaría a escribir un software capaz de modificar nuestras trayectorias expresadas en modo cartesiano a otras adaptadas a nuestro plóter. Además, pequeños giros del motor tienen gran influencia en el movimiento de la estructura, por lo que la precisión no será particularmente alta.

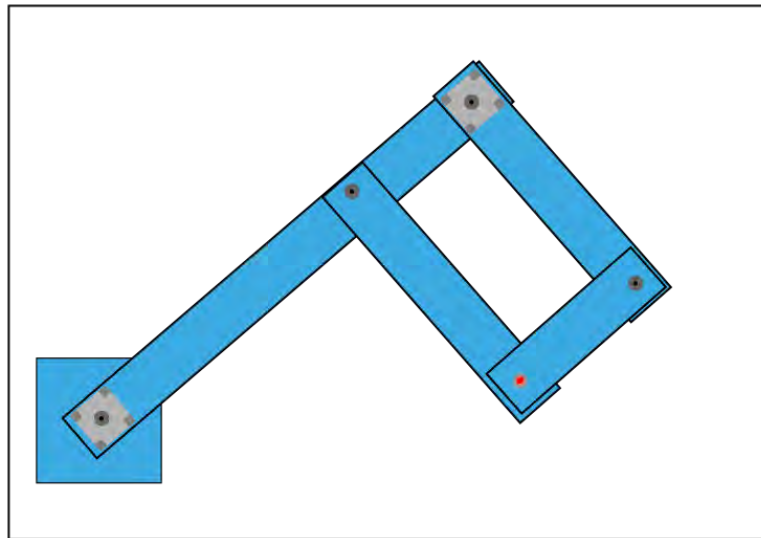


Figura 3.2: Ploter pantografo

3.2.3. Plóter portable con estructura de paralex

Un sistema relativamente portátil podría ser realizar el movimiento en el eje Y con un carro fijo y en el eje X con una estructura de paralex. Uno de sus mayores defectos es que sigue dependiendo de hilos para moverse en un eje, lo que disminuye su precisión. Además, el movimiento del carro sobre el área de trabajo se produce por deslizamiento, y queremos evitar en la medida de lo posible que exista contacto entre estos dos elementos. Si bien esto se puede solucionar utilizando cuatro canicas en los extremos, la falta de precisión en un eje es un problema de difícil solución.

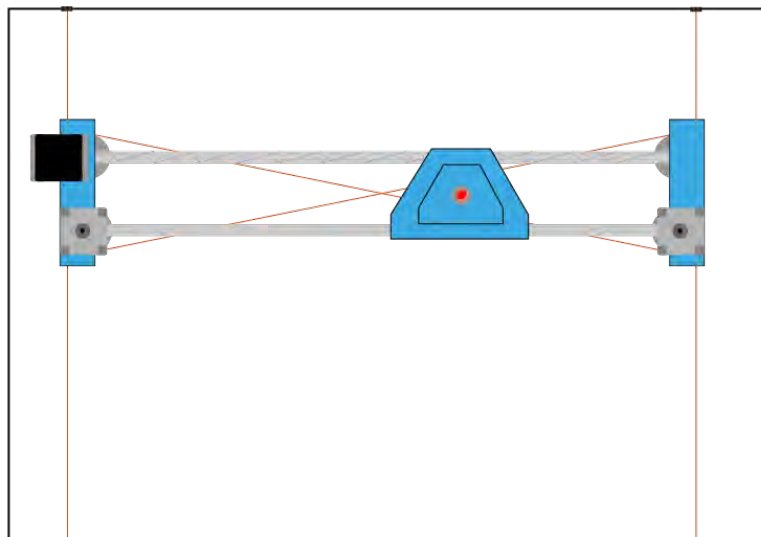


Figura 3.3: Ploter paralex

3.2.4. Plóter fijo X-Y

El sistema más utilizado para trazar trayectorias en el plano, es sencillo y funcional. Tiene la ventaja de ser extensible con facilidad. Probablemente uno de los mayores defectos que tenga es que es de dimensiones fijas y que a partir de ciertas dimensiones no es nada portable. Por sencillez y comodidad en su desarrollo se ha optado por este diseño para realizar el prototipo.

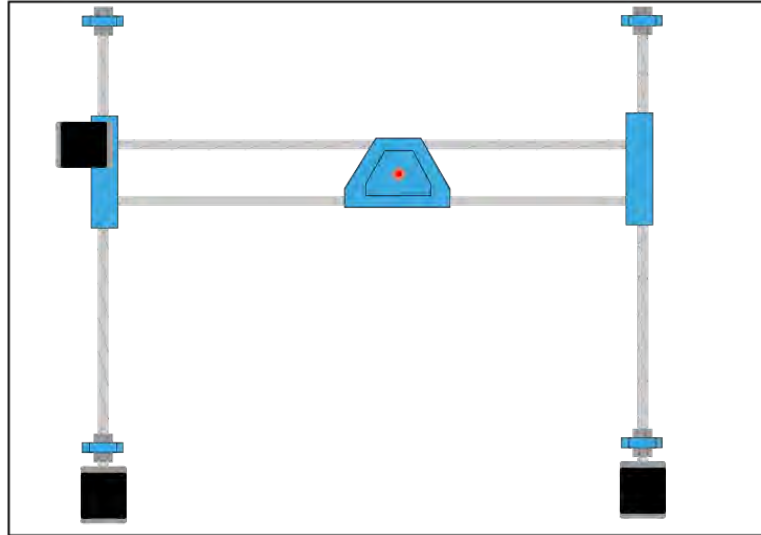


Figura 3.4: Ploter X-Y fijo

3.2.5. Plóter X-Y portable

Como se puede apreciar esta propuesta es prácticamente idéntica al plóter fijo X-Y con la salvedad de que esta estructura se divide en partes cómodas de transportar. Tiene sentido cuando pensamos en un plóter de grandes dimensiones (Por ejemplo 1m^2).

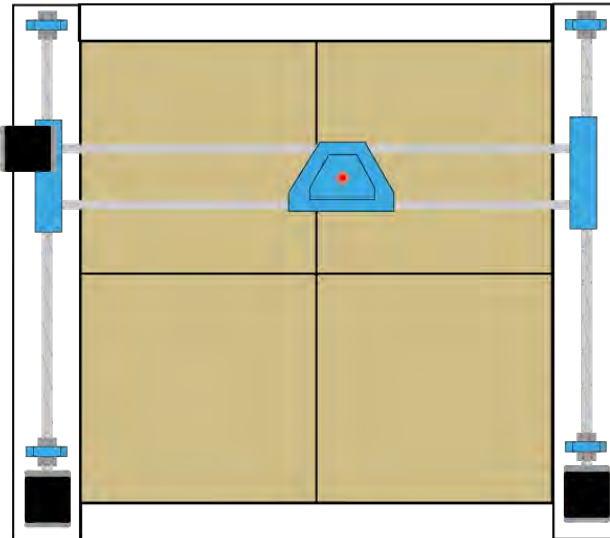


Figura 3.5: Plóter X-Y portable

3.3. Elecciones de hardware, software y modo de fabricación

Nuestro sistema de plóter se compone de la electrónica de control, los motores, la mecánica que va a determinar el movimiento, compuesta por piezas comerciales y piezas de diseño propio, las herramientas y el servidor web con su correspondiente interfaz.

A la hora de diseñar Proteus, es importante recordar que los objetivos fundamentales son hacerlo accesible para la comunidad, tanto a nivel de documentación como económicamente, hacerlo multiherramienta y lo más preciso posible. Con las elecciones explicadas en este apartado se ha buscado conseguir un equilibrio entre estos objetivos.

3.3.1. Elección del controlador

Para cumplir con los requisitos propuestos para este proyecto era necesario un controlador capaz de procesar las trayectorias definidas en nuestro software, controlar los motores, las herramientas y que pueda actuar de servidor para la aplicación de control. Si bien es cierto que para una producción en serie puede resultar interesante diseñar una electrónica propia adaptada a las necesidades específicas de este proyecto, a la hora de realizar un prototipo, y más si tenemos el objetivo de poner este al servicio de la comunidad maker, es más interesante (por económico y por facilidad de acceso a la electrónica) utilizar placas comerciales. Se plantean varias posibles soluciones:

- Una Raspberry Pi para actuar como servidor y controlador de los motores y herramientas mediante un shield específico y los drivers pololu. Dado que el control se realizaría mediante los puertos GPIO y que el procesador debería atender al sistema operativo además de al control, esta opción parece ineficiente frente a las demás. Por otra parte el shield específico es caro, por lo que esta solución no nos aporta ventajas claras.
- Un Arduino Yun que actúe como servidor y controlador de motores y herramientas mediante un shield y los drivers pololu. Este Arduino tiene en realidad un procesador de ordenador que gestiona un sistema operativo linux (Un AtherosAR9331) y un procesador de microcontrolador similar a un Arduino UNO (Concretamente un AtMega32U4)
- Una Raspberry Pi para actuar como servidor y un Arduino UNO con un shield y los drivers pololu para el control de los motores y las herramientas. Es la más económica de las posibles soluciones, y además tiene un procesador dedicado al control de Proteus y otro al sistema operativo que mantiene levantado el servidor. Si bien se podría utilizar otro Arduino, como el DUE, el Mega o el Leonardo, el Arduino UNO tiene una excelente relación características/precio y es suficiente para nuestros propósitos. Por estas razones esta ha sido la electrónica elegida para desarrollar este proyecto.

3.3.2. Elección de la integración de las herramientas

A la hora de decidir crear un plóter multiherramienta, es una decisión fundamental decidir como se integrarán estas con la máquina que produce propiamente el movimiento y de que forma se realizará su control con sencillez y eficacia. Por este motivo se decidió que las herramientas serían un binomio: por una parte la herramienta propiamente dicha, que se integra con Proteus mediante el diseño de cabezales fácilmente intercambiables, y por otra la parte del control electrónico, el driver, que se integra con la electrónica general del sistema basando su estructura en los drivers de los motores paso a paso pololu y utilizando las mismas señales para ser comandados. De esta forma, la sustitución de una herramienta por otra se puede realizar de forma sencilla sin requerir de cambios en la configuración del software.

3.3.3. Elemento tractor: Correas o varillas roscadas

Entre las elecciones en el diseño de la estructura, se encuentra la elección de la mecánica que va a producir el movimiento, que puede realizarse con correas o varillas.

El movimiento con correas requiere correas, ruedas dentadas, varillas lisas y rodamientos lineales. Esta mecánica permitirá movimientos rápidos, pero por otra parte el engrane entre la correa y la rueda dentada y la distancia recorrida por vuelta del motor ($2 * \pi * R$, siendo R el radio de la rueda dentada) afectarán de forma negativa a la precisión de nuestros movimientos.

El movimiento con varillas roscadas requiere la varilla roscada, una tuerca y que nuestra estructura compense la fuerza de torsión dejando fija la rotación de la tuerca en los dos ejes que no son el principal. De esta forma, con cada vuelta del motor se avanza poco. Esta mecánica será lenta, pero por la prácticamente inexistente holgura entre la tuerca y la varilla y su menor desplazamiento por vuelta del motor, la precisión será mucho mayor.

Debido a lo anteriormente comentado y teniendo en cuenta que conseguir la mayor precisión posible es una de las metas fundamentales de este proyecto, se optó por las varillas roscadas, ya que la baja velocidad no supone un grave problema, pues las herramientas se pueden adaptar a trabajar a menores velocidades.

Medidas

Dado que la estructura de Proteus se compone de una serie de piezas estándar y otras de diseño propio, el diseño de estas últimas se planteó de tal manera que incrementar o reducir su tamaño sea lo más sencillo posible. Esto se ha conseguido determinando las medidas del plóter (y por tanto de su área de trabajo) únicamente con las medidas de las varillas metálicas. Experimentalmente se ha comprobado que los motores que utilizaremos, NEMA 17, mueven satisfactoriamente una varilla de un metro de longitud, pero dado que queremos que el plóter sea portable, trabajaremos con varillas de 50 cm. Entre las distintas soluciones posibles se encuentra un plóter portable de mayores dimensiones. Una serie de ideas sobre como podría realizarse esto se explica en el apartado de futuros desarrollos.

3.3.4. Aplicaciones para smartphones, tablets y ordenadores

A día de hoy existen en el mundo más teléfonos móviles que personas y se estima que el 15 % de la población mundial usa un tablet. Entre Android e iOS tienen cerca del 95 % del mercado de dispositivos portátiles. Estos dispositivos tienen acceso a infinidad de aplicaciones y en la mayoría de los casos, conexión a internet.

Cuando se diseña una aplicación móvil existen fundamentalmente dos opciones: O se diseña una aplicación compatible con un sistema operativo móvil específico (Android, iOS, WindowsPhone, Tizen...) de manera que para ofrecerla al público es necesario contar con la aprobación de las distintas empresas (Google, Apple, Windows, Nokia). O por otra parte se puede diseñar una aplicación web a la que se pueda acceder desde cualquier navegador compatible con la tecnología que utilizada. La forma más cómoda de realizar una aplicación multiplataforma e independiente del sistema operativo del dispositivo que se vaya a utilizar es usar HTML5 y CSS3. Si usamos JavaScript deberá ejecutarse del lado del servidor. Por este motivo utilizaremos un servidor de node.js haciendo la aplicación visualizable desde smartphones, tablets o pc's.

3.3.5. Interfaz

Si bien la interfaz web es la del proyecto open source utilizado para controlar el plóter, se han realizado algunas modificaciones que mejoran la seguridad al utilizar Proteus. Concretamente se ha creado un control de acceso mediante un servidor de node.js que conduce a una pantalla de login para acceder a la aplicación de control de Proteus desde cualquier dispositivo. Estas mejoras y la documentación generada sobre ellas serán aportadas a la comunidad para que cualquiera pueda implementarlas y utilizarlas.

3.3.6. Elección del modo de fabricación

Como hemos comentado anteriormente, la estructura de Proteus se compone de piezas estándar compradas (varillas, tornillos, rodamientos, tuercas...) y piezas de diseño propio. Para fabricar estas últimas, se han estudiado diversas soluciones de fabricación para la estructura tales como el mecanizado CNC o el moldeo en arena, pero se ha optado por la impresión 3D dada la facilidad y flexibilidad que nos da a la hora de hacer prototipos rápidos a bajo coste, permitiéndonos iterar nuestros diseños hasta conseguir las piezas deseadas. Concretamente se ha utilizado una MakerBot Replicator 2X. El material utilizado es Acrilonitrilo Butadieno Estireno (ABS) que es económico (Aproximadamente 15 euros/Kilo), resistente y ligero.

Aunque este modelo de fabricación es muy interesante por su flexibilidad, para producción en serie resultaría más interesante fabricar las piezas por inyección por economía de escala.

Capítulo 4

Descripción del Hardware

En este capítulo se va a describir el hardware utilizado en el desarrollo de Proteus. Trataremos el hardware que compone la unidad de control, los motores que se encargan de todo el movimiento del plóter y el que compone el ordenador que será servidor de la webapp (Y se comunicará con la unidad de control para realizar los trazos pertinentes) y las distintas fuentes de alimentación que requiere el hardware. El hardware específico de las distintas herramientas será descrito en capítulos posteriores.

4.1. Unidad de Control

La unidad de control de Proteus, encargada de mover el plóter, está compuesta por tres elementos:

- Un Arduino UNO que es la placa micro-controladora.
- Un CNC Shield que hace de interfaz entre el Arduino y los pololu.
- Tres drivers Pololu A4988 que se encargan de mover los motores.

4.1.1. Arduino Uno

Arduino es un entorno de desarrollo compatible con toda una familia de placas micro-controladoras (UNO,DUE, Yún, Leonardo...) ideales para variedad de proyectos dada su flexibilidad y economía. Estas placas micro-controladoras utilizan generalmente microprocesadores Atmel, concretamente los ATmega328, ATmega1280 y ATmega2560. Se programan en processing, que es un lenguaje basado en C++.

Para el desarrollo de Proteus se ha elegido trabajar con un Arduino UNO por ser una placa con mucha documentación generada por la comunidad y por ser de muy bajo coste [6]. Esta placa consta de un microprocesador ATmega328 y se puede alimentar desde la conexión USB. Consta de seis entradas analógicas y catorce pines digitales de Entrada/Salida, de las cuales seis pueden proporcionar PWM.



Figura 4.1: Placa microcontroladora Arduino UNO.

El microcontrolador Atmega328 que tiene el Arduino UNO opera a un voltaje de 5V y a una frecuencia de 16 MHZ, posee una memoria flash de 32KB (0.5KB ocupados por el bootloader), SRAM 2 KB, y EEPROM 1KB [7].

4.1.2. CNC Shield

La CNC Shield es una placa que se conecta a Arduino y dispone los pines del Arduino Uno de forma que tengamos acceso de forma cómoda y sencilla a las distintas funciones de un CNC que nos da GRBL. Es una placa Open Source y Open Hardware totalmente compatible con GRBL [8].

Dispone de cuatro socket, que incluyen un condensador conectado, para los drivers de los motores, para la herramienta de trabajo (Motor 'A') y para los que mueven los ejes X, Y y Z. Esta placa permite configurar si queremos que el motor 'A' tenga los pasos y dirección de otro de los motores. En nuestro caso, los dos motores que mueven el eje Y deben comportarse igual, y por lo tanto, el driver del motor A copiará dirección y pasos al driver del eje Y.

Consta de seis conexiones para los sensores finales de carrera, dos por eje, de manera que podremos detectar el inicio y final de cada varilla para no forzar el movimiento una vez terminado el recorrido.

Además también dispone de conexión para ventilador y conexión para botones de cancelar, parar y continuar.

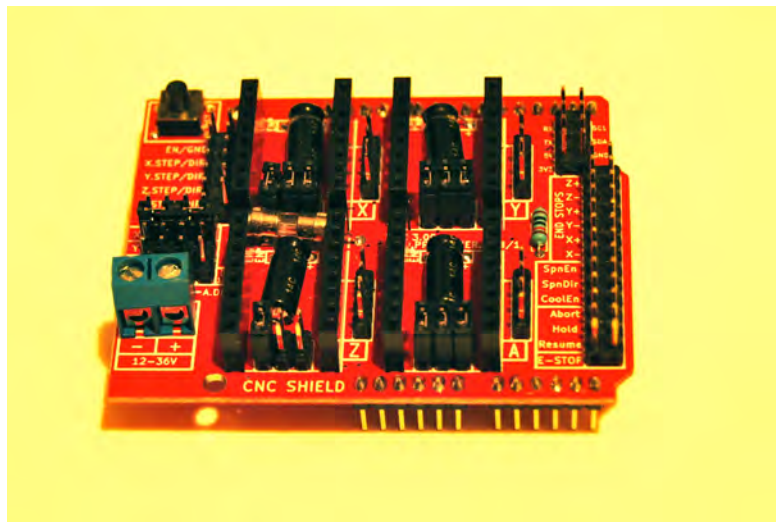


Figura 4.2: CNC Shield.

4.1.2.1. Drivers pololu

Esta placa utiliza el driver A4988 (Allegro) que tiene limitación ajustable de corriente, protección contra subidas de corriente y cinco resoluciones de microstepping. Esto permite controlar los motores paso a paso NEMA 17 con una precisión de hasta 1/16 de paso [10].

- Voltaje de funcionamiento: 8-35 V
- Corriente de salida: 1A continuo (2 A con disipador)
- Dimensiones: 20,3x15,2mm

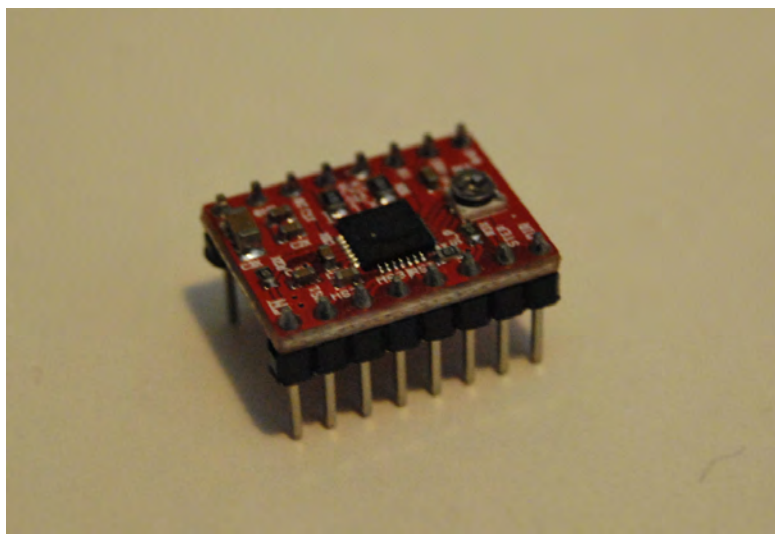


Figura 4.3: Driver A4988 de motor paso a paso.

4.2. Motores NEMA 17

Los motores NEMA 17 son motores paso a paso con unas dimensiones de 42.3mm x 42.3mm de cara y longitud según modelo. Se han elegido unos NEMA17 bastante económicos, de 10 €/motor. Las características específicas de los motores elegidos son:

- Bipolar
- El ángulo de cada paso es de 1.8° (200 pasos por vuelta)
- La longitud del cuerpo es de 40mm.
- Corriente máxima de 1.7A
- Resistencia de 20hm
- Inductancia de 3 mH
- Peso: 300g
- Diámetro del eje: 5 mm
- Longitud del eje: 25 mm
- Torque 2.5Kg/cm



Figura 4.4: Motor paso a paso NEMA17.

4.3. Fuente de alimentación

La fuente de alimentación de Proteus es una fuente de alimentación de PC, concretamente una ATX modificada, cuya función es alimentar la CNC Shield. Estas fuentes constan de conectores de 24 pines. Para poder utilizarla sin estar conectada a la placa base de un ordenador, cortocircuitaremos el pin PS ON con GND. En nuestro caso, cortocircuitaremos el pin 16 con el 15 insertando medio clip. Además, juntaremos varios cables negros y varios cables amarillos y los uniremos a una clema para tener la alimentación de nuestra CNC Shield (+12 V, GND). Sería posible sacar alimentación para la Raspberry Pi, pero queremos mantener la independencia del plóter y del servidor de la aplicación de control.



Figura 4.5: Fuente de alimentación

1	2	3	4	5	6	7	8	9	10	11	12
+3.3 V	+3.3 V	GND	+5 V	GND	+5 V	GND	Power OK	+5 VSB	+12 V	+12 V	+3.3 V
+3.3 V	-12 V	GND	PS ON	GND	GND	GND	-5 V	+5 V	+5 V	+5 V	GND
13	14	15	16	17	18	19	20	21	22	23	24

Tabla 4.1: Conexiones de Fuente ATX

4.4. Miniordenador con el servidor de la webapp: Raspberry Pi

La Raspberry Pi Foundation es una organización benéfica educativa cuyo objetivo es promover entre niños y adultos una educación avanzada en el campo de la informática [9].

Con este fin se creó el miniordenador denominado Raspberry Pi, del cual hay varias versiones:

- Raspberry Pi A/B: Las primeras en ser presentadas (Febrero de 2012). Con un ARM6 de 700MHz y 256MB de RAM solo se diferencian en que el modelo B tiene 2 puertos USB, uno más que el modelo A, y conexión ethernet. Posteriormente comenzó a fabricarse con 512 MB de RAM.
- Raspberry Pi A+/B+: Modelos con algunas mejoras de eficiencia energética sobre sus predecesoras y un añadido interesante en el modelo B, 4 puertos USB.
- Raspberry Pi 2 B: Es el modelo más reciente, con muchas mejoras, entre las que se encuentra un procesador ARM7 de cuatro núcleos y 1GB de RAM.

En todas las generaciones, los modelos A han costado entre 20 y 25 dólares y los modelos B 35 dólares. Por ello se ha optado por el modelo B. Aunque es algo antiguo, es fácil de conseguir por unos 20 euros, y dado que ofrece todas las prestaciones que en principio se requieren para ejecutar el software GRBLWeb, supone un ahorro interesante en nuestro presupuesto.

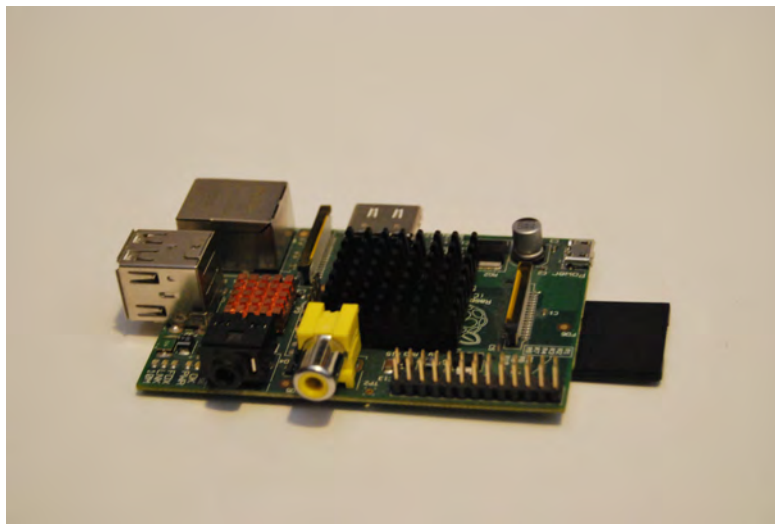


Figura 4.6: Miniordenador Raspberry Pi B modelo de 256Mb.

4.4.1. USB Wifi

Dado que la Raspberry Pi no tiene Wifi integrada, es necesario utilizar un adaptador USB que la dota de esta capacidad, de manera que podemos acceder a internet con la Raspberry para instalar software y además, esto nos permite configurar el acceso al servidor web de la aplicación vía Wifi.



Figura 4.7: Adaptador USB Wifi

4.5. Transformador

La función del transformador es alimentar nuestra Raspberry Pi, que es el servidor de nuestra aplicación, a través de su puerto microUSB. Para ello debe convertir los 220V AC de la red eléctrica a 5V.

Las especificaciones concretas del cargador son:

- Input: 100-240VAC 50/60Hz 0.3A
- Output: 5V - 1200mA



Figura 4.8: Transformador de la Raspberry Pi.

Capítulo 5

Proteus

El plóter Proteus, que toma su nombre del dios griego metamórfico por su orientación multiherramienta, se compone fundamentalmente de tres partes: la electrónica, que se ha detallado en el capítulo de descripción del hardware y en este se verá como queda conexionada, las piezas estándar de ferretería que hacen de este plóter una máquina fácilmente replicable y por supuesto, las piezas impresas específicamente diseñadas para conseguir que el plóter sea una máquina versátil y económica.

5.1. Piezas Impresas

Las piezas se han impreso en una Makerbot Replicator 2X, su diseño es libre y han sido diseñadas teniendo en mente una serie de objetivos muy concretos:

- Minimizar el gasto de plástico sin afectar a la robustez del diseño.
- Que permitiese variar sus dimensiones de forma sencilla.
- Que las piezas tuviesen tiempos de impresión que no fuesen superiores a una hora.
- Que no hubiese zonas con ángulos difíciles para las impresoras 3D.

5.1.1. Diseño: OpenSCAD

Se ha escogido este software porque, al ser las piezas de código abierto para permitir que quien quiera pueda utilizar estos diseños, parecía razonable utilizar un lenguaje que fuese descriptivo y fácil de entender, además de ser muy sencillo de manipular, lo que permite comenzar a trabajar con ello sin una serie de conocimientos técnicos que pueden ser necesarios para utilizar otros programas.

5.1.2. Descripción de las piezas

La estructura de Proteus se compone de varias piezas impresas junto con varilla, tornillos y tuercas. Estas son:

1. **Cojinetes para los rodamientos (6):** Esta pieza es la que permite rotar a las varillas roscadas, pues son unos soportes para los rodamientos radiales. Son además parte del apoyo de la estructura.



Figura 5.1: Cojinetes impresos para los rodamientos radiales

2. **Soporte superior de los motores (2):** Es una pieza que fija los motores a la base, alineados con la varilla roscada a la que van unidos. Esta pieza va atornillada a la base y al motor.

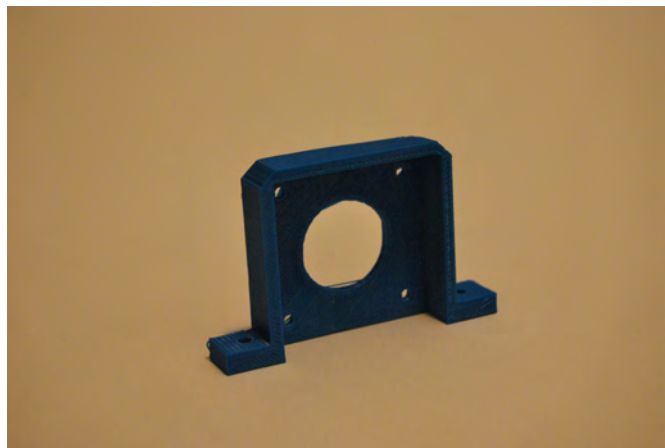


Figura 5.2: Anclaje superior de los motores.

3. **Soporte inferior del motor derecho (1):** Esta pieza en conjunto con la anterior anclan el motor, evitando que este se mueva y constituyen parte del apoyo de la estructura del plóter.

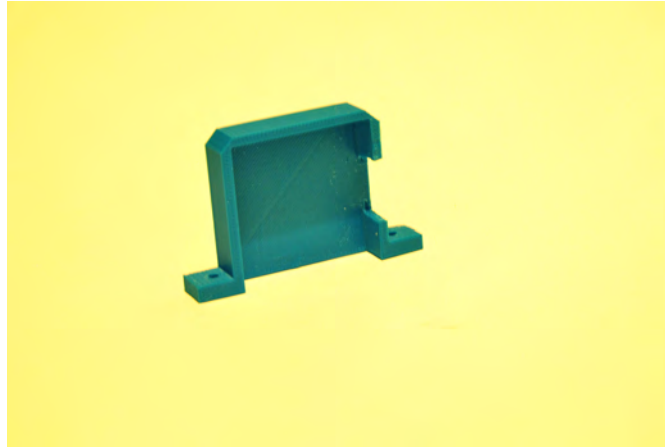


Figura 5.3: Anclaje inferior del motor derecho.

4. **Soporte inferior del motor izquierdo (1):** Prácticamente idéntica a la pieza anterior, pero diseñada para el motor izquierdo, solo se diferencian en donde se encuentra ubicado el espacio para las conexiones eléctricas de los motores.

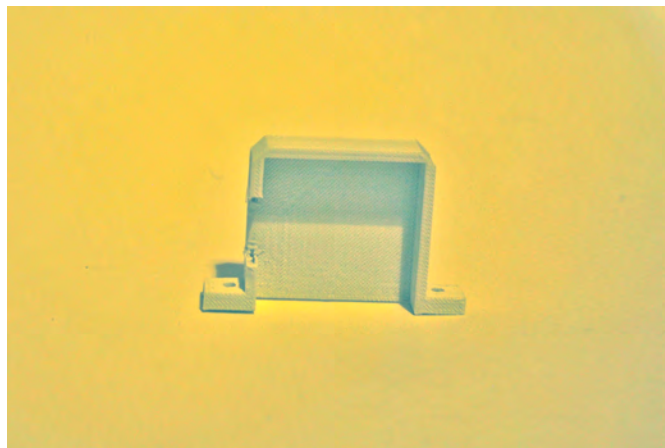


Figura 5.4: Anclaje inferior del motor izquierdo

5. **Unión motor-varilla superior (2):** El movimiento de los motores debe transmitirse a las varillas de forma solidaria, para ello se han diseñado una pareja de piezas que al atornillarse la una a la otra abrazan tanto a la varilla como al eje del motor, para realizar un mejor ajuste con el eje se ha utilizado aproximadamente un centímetro de tubo de plástico transparente.



Figura 5.5: Parte superior de la unión del motor con la varilla roscada

6. **Unión motor-varilla inferior (2):** Pareja de la pieza anterior. En un futuro rediseño, se ha planeado hacer que ambas partes sen idénticas, de manera que no se puedan cometer errores como utilizar dos caras inferiores para realizar una unión.



Figura 5.6: Parte inferior de la unión del motor con la varilla roscada

7. **Soporte izquierdo de la deslizadera (1):** Esta pieza es particularmente importante por varios motivos: contiene la tuerca fija en su interior que hace que la rotación de la varilla roscada en un sentido o en otro produzca el desplazamiento del carro de la herramienta, junto con la pieza derecha y las varillas correspondientes conforman la deslizadera y soporta el motor del eje X que va a producir el movimiento del carro de la herramienta.

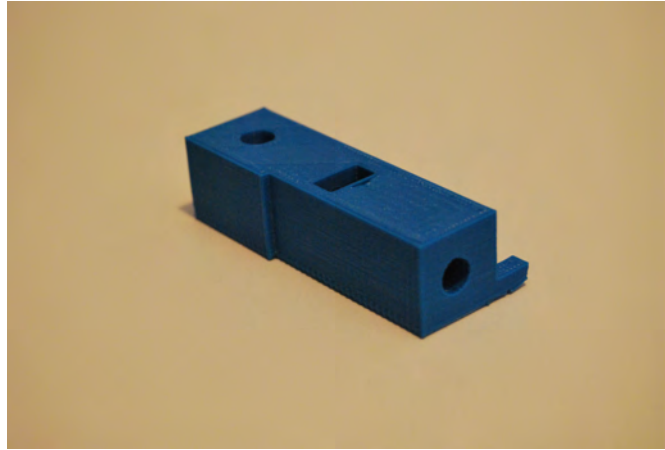


Figura 5.7: Parte izquierda de la deslizadera.

8. **Soporte derecho de la deslizadera (1):** Muy similar a la anterior no tiene que soportar el motor sino que solo soporta el cojinete de la varilla roscada del eje X.

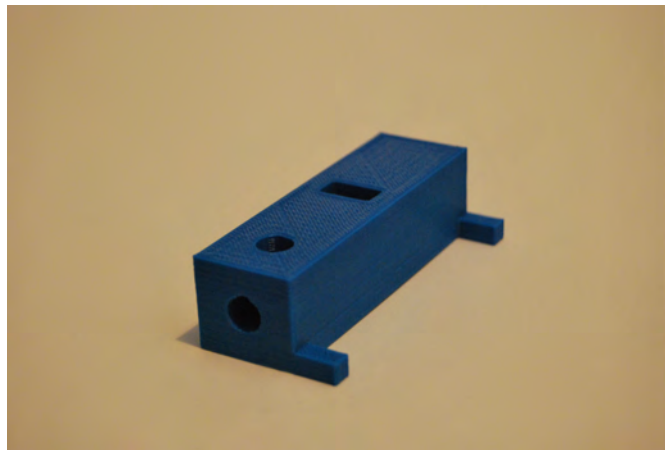


Figura 5.8: Parte derecha de la deslizadera.

9. **Carro de la herramienta (1):** El carro de la herramienta es la pieza que transporta las distintas herramientas mediante una pieza de unión específica en cada herramienta.

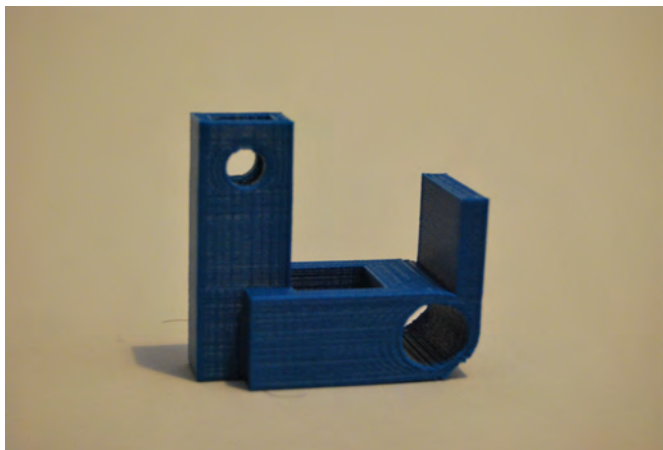


Figura 5.9: Carro de la deslizadera

Además de estas piezas, existe una pieza específica de cada herramienta que se explicará en su correspondiente capítulo.

Estas piezas se pueden modificar para aceptar varillas de otras dimensiones, por lo que se puede adaptar a las dimensiones estándar de cada país. Cabe mencionar que el diseño de todas las piezas se ha visto afectado por el modo de fabricación que limita las formas de diseño.

Nota: Los códigos de las piezas diseñadas en OpenSCAD se encuentran en el apéndice [B](#).

5.1.3. Fallos de impresión y mejoras aplicadas

Tras realizar las primeras pruebas de impresión se descubrió que la impresora realizaba las piezas casi medio milímetro más grandes que los diseños originales, así que se rediseño la pieza en cuestión (las uniones motor-varilla) y se consideró este factor en los diseños sucesivos.

También se observó en algunas piezas que los diseños originales no encajaban perfectamente, o en el caso más notable (el de los cojinetes), no aportaban la solidez estructural deseada, por lo que se rediseñaron en varias ocasiones algunas de las piezas hasta que se ajustaron a lo que el plóter requería.



Figura 5.10: Cambio de diseño en los cojinetes

5.2. Piezas estándar de ferretería

Para unir entre si las piezas impresas de Proteus y crear su estructura estructura se han utilizado unas pocas piezas estándar muy fáciles de encontrar. Concretamente se han utilizado:

- Varillas roscadas de 8 mm: Se encuentran conectadas a los motores. Son las encargadas de transmitir el movimiento.
- Varillas lisas de 8 mm: Se usa una varilla lisa en la deslizadera, por ella se desplaza el carro de la herramienta deslizando.
- Tuercas de 8 mm: Son las receptoras del movimiento de rotación que transmiten las varillas roscadas. Al estar fijas y no poder rotar, transforman este giro en un desplazamiento lineal.
- Rodamientos lineales de 8 mm: Se usan en el carro de las herramientas, y son las piezas que deslizan sobre la varilla lisa.
- Rodamientos radiales de 8 mm: Se usan para hacer rotar las varillas roscadas con el mínimo rozamiento posible. Se utiliza un rodamiento en cada extremo de las varillas.
- Tornillos de 3 mm: Utilizados para anclar las piezas de apoyo de la estructura a la base en que se sustentan.
- Tuercas de 3 mm: Se utilizan para fijar los tornillos mencionados anteriormente.

5.3. Esquema de montaje de la electrónica

Una vez contamos con todas las piezas de hardware descritas, se deben montar y conectar unas con otras. Este es un proceso sencillo pero debe realizarse cuidadosamente para no cometer errores que puedan afectar posteriormente al funcionamiento del plóter.

Lo primero sería conectar en Arduino UNO con la CNC Shield y sobre esta colocar los jumpers que permitan replicar el motor utilizado para el eje Y en el socket A (de manera que ambos motores se moverán de la misma forma), colocar un driver pololu A 4988 en cada uno de estos dos socket y, una vez conectada la fuente de alimentación, calibrar la intensidad de los drivers.

Para calibrar la intensidad que circula por los drivers debemos mirar la datasheet del driver [11]. La corriente máxima para el driver es:

$$I_{max} = V_{ref} / (8 * R_s)$$

Para calcular esto se medirá la tensión entre V_{ref} y GND . Como la resistencia de los drivers según el modelo puede ser $R_s = 0,05$, $R_s = 0,1$ o $R_s = 0,2$ es importante cerciorarse del valor de las resistencias del driver de que se disponga.

Ponemos un valor de menos de un amperio a la intensidad que circulará por el driver, (por ejemplo 0.8) y calculamos V_{ref} en función de nuestra R_s . Reordenando la fórmula anterior obtenemos:

$$V_{ref} = I_{max} * 8 * R_s$$

Giramos el potenciómetro en sentido horario para aumentar la intensidad y en sentido antihorario para disminuirla hasta obtener el valor de V_{ref} calculado.

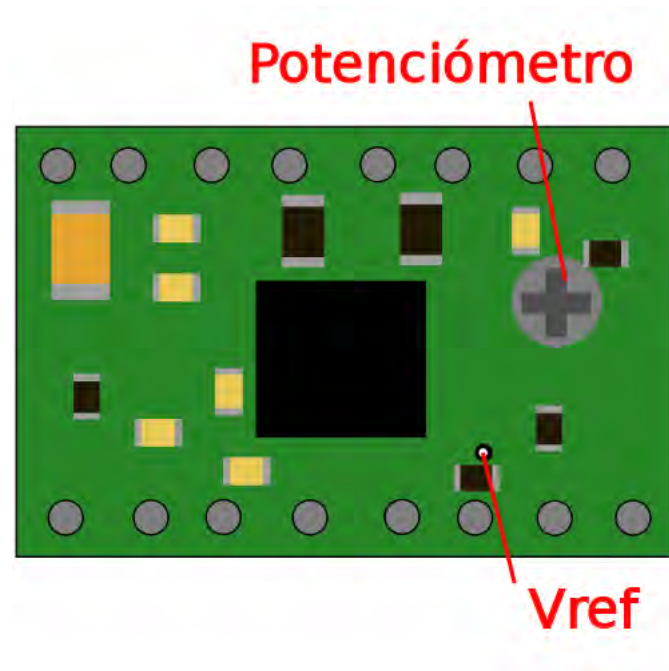


Figura 5.11: Potenciómetro y V_{ref} para calibrar la intensidad que circula por el driver

Luego conectamos el driver del tercer motor, los sensores finales de carrera, el botón de interrupciones y el driver de la herramienta a utilizar. Conectamos por USB el Arduino UNO a la Raspberry Pi y esta a la alimentación correspondiente. Por tanto, el circuito completo quedará así:

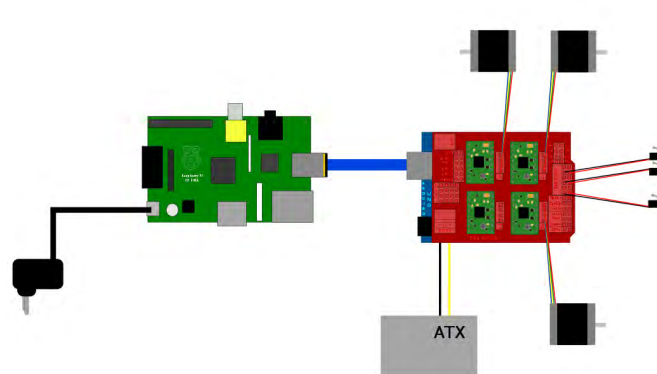


Figura 5.12: Esquema con las conexiones de toda la electrónica

Este circuito se presenta a mayor resolución en el Anexo 3

Capítulo 6

Módulo de herramienta: Surya

El módulo Surya, llamado así por el dios hindú del sol, es el módulo de herramienta que utiliza un diodo láser para cortar y grabar distintos materiales. Los diodos láser rojos de 650nm y 250mA pueden grabar madera, plástico, ... así como cortar papel, cartón, vinilo, y tejidos finos. Todas estas posibilidades hacen de este modulo una gran herramienta, de la cual se detallarán posibles usos al final del capítulo

6.1. Hardware específico

El hardware específico que vamos a utilizar va a ser un diodo láser con un disipador y un driver de diseño propio.

6.1.1. Láser

Un diodo láser, al polarizarse, emite un haz de luz coherente. Son dispositivos eficientes y fiables que se utilizan desde la industria médica hasta la fabricación de lectores/grabadores de DVD o impresoras láser.

Características técnicas:

- Potencia: 250mW
- Longitud de onda: 650nm, láser rojo
- Alimentación: 3-5V CC
- Tiempo de vida: superior a 6000 horas

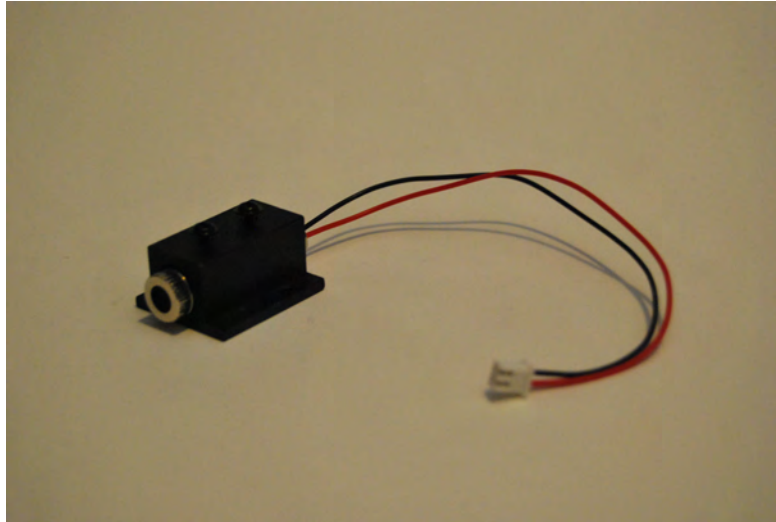


Figura 6.1: Láser

6.1.2. Driver

Para el uso del diodo led en nuestra herramienta, necesitamos poder regular la potencia del láser, su encendido y su apagado. Para conseguir esto, y dado que la mayoría de diseños de drivers para láser constan de un regulador de tensión LM317 que junto con un potenciómetro regula la intensidad del circuito, será suficiente con conseguir que nuestra señal de control DIR encienda o apague el láser en función de si es 0 o 1.

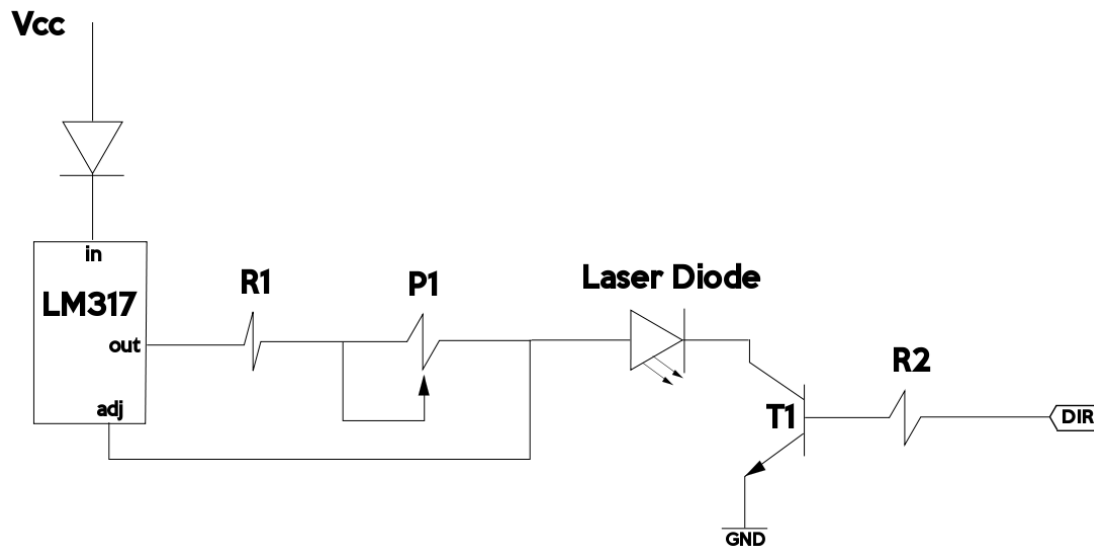


Figura 6.2: Driver para láser controlado

Para nuestro diseño es conveniente asociar este diseño de driver controlado para láser al formato de pines del driver pololu A4988. Alimentamos el diodo láser con la alimentación de los motores, asociamos el GND de nuestro driver al GND de la placa y utilizamos como control la señal de DIR sobre la base del transistor.

De esta forma lo que conseguimos es que cuando tenemos DIR=0 la tensión en base es 0 y el circuito está interrumpido y cuando tenemos DIR=1 el transistor cierra el circuito activando el diodo láser. Al conectar el láser se debe recordar que el cable rojo es el ánodo y el negro el cátodo. Conectar erróneamente el láser hará que este no funcione, e incluso podría dañarse.

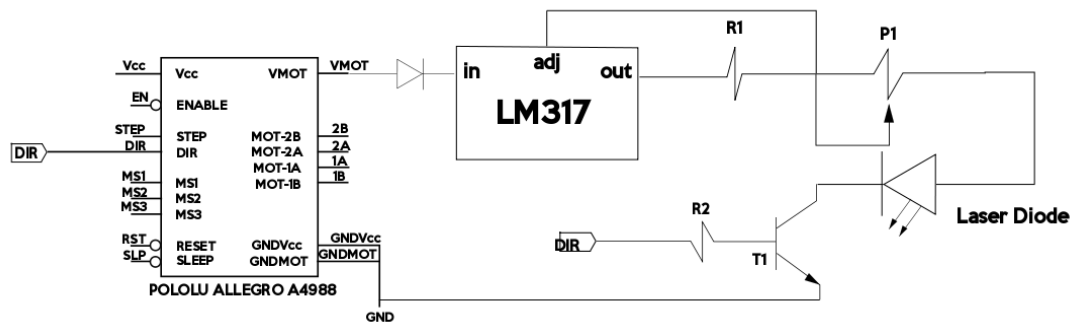


Figura 6.3: Driver para láser controlado asociado al formato pololu

Al ser un prototipo, se ha diseñado el driver sobre una placa de prototipado rápido manteniendo las dimensiones y pines del driver pololu para motores paso a paso. Si finalmente se comercializase en forma de kit, lo más apropiado sería diseñar una pcb para esta herramienta que sea más robusta y segura.

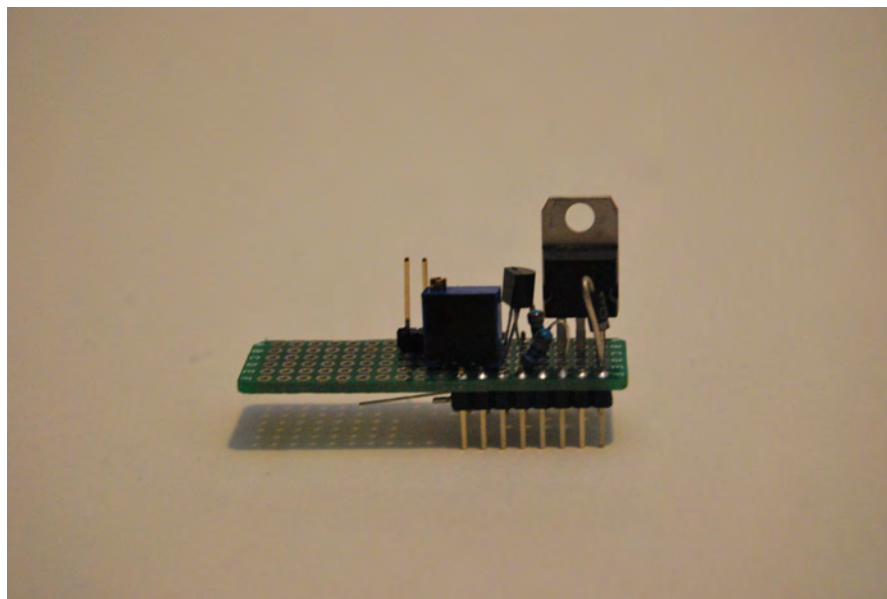


Figura 6.4: Driver para láser montado

6.2. Pieza de agarre al carro de la deslizadera

Esta pieza está atornillada al láser y se fija al carro de la deslizadera cuando se usa la herramienta Surya. Así el láser queda fijado sobre el hueco del carro y puede cortar/grabar el material que tiene debajo.

El código de OpenSCAD de esta herramienta se encuentra con el resto de piezas impresas en el apéndice [B](#)

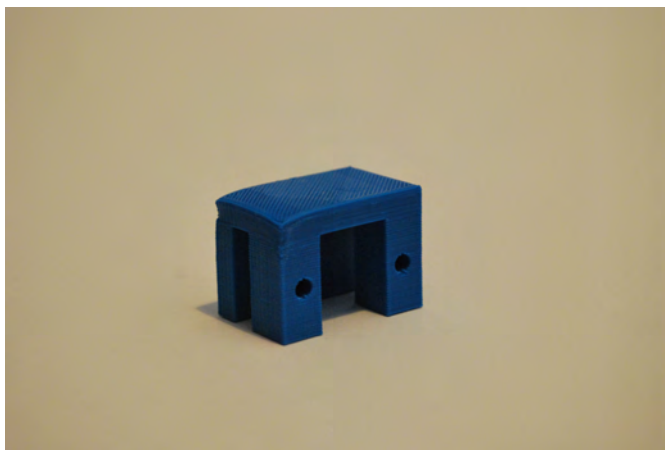


Figura 6.5: Pieza de agarre al carro de Surya

6.3. Seguridad

Cuando el plóter esté funcionando con esta herramienta, debemos tener en cuenta que el diodo láser puede dañar de forma permanente nuestra vista y por ende es aconsejable seguir algunas medidas básicas de seguridad:

- Nunca se debe encender el láser, o permanecer en la habitación en la que esté el plóter configurado con esta herramienta sin disponer de las gafas de seguridad correspondientes. El diodo láser no es un juguete y su uso incorrecto puede producir daños en la visión e incluso ceguera. Por supuesto, el diodo láser también puede producir quemaduras e incendios, por lo que tampoco es recomendable dejar la herramienta totalmente desatendida.
- Mientras la máquina esté en funcionamiento no se debe intentar acceder a la mecánica de la misma, ni para colocar algo que no esté en la posición que pensamos que es correcta ni para ninguna otra cuestión. Si se debe hacer cualquier ajuste, se debe parar el plóter, realizar el ajuste y reanudar la marcha del mismo.
- No se debe tocar la electrónica, desconectar o conectar drivers, ajustar los cables de la alimentación, etc... con la fuente de alimentación encendida. Mejor apagarla, realizar las conexiones y comprobaciones pertinentes y posteriormente encenderla.

- Esta herramienta funciona controlando un láser, que se proyecta sobre el material de trabajo y lo calienta hasta ser capaz de realizar grabados mediante quemado o incluso cortar el material. No se debe poner nunca nada entre medias del láser y el material. Hacer esto puede resultar en quemaduras e incluso incendios, poniendo en peligro todo el inmueble donde se encuentre.

6.4. Posibles usos

Esta herramienta tiene una orientación mayormente técnica, aunque por supuesto se puede utilizar con fines creativos y artísticos. Permite realizar grabado y corte de trazos diseñados por CAD, lo que tiene enormes posibilidades:

6.4.1. Creación de PCBS

Uno de los posibles usos más interesantes de esta herramienta es la fabricación de PCBS caseras de gran calidad. Para esto se debe dejar el diseño de la PCB en blanco y negro, donde las pistas deben estar en blanco. Luego se pinta con pintura en spray negra una placa de cobre para hacer PCBS y sobre ella se graba nuestro diseño. Al hacer esto, se quema la pintura, que se despegas de la placa de cobre. Se limpia la superficie con un bastoncillo impregnado de alcohol etílico con cuidado de no dañar las zonas que se quieren mantener protegidas y se obtiene la PCB lista para un baño de cloruro férrico en el que se eliminarán las zonas desprotegidas, dejando, tras limpiarse la placa con un algodón con alcohol etílico una PCB lista para taladrar y utilizar.



Figura 6.6: Creación de PCBS [15].

6.4.2. Personalización de objetos

Los objetos plásticos o de madera se pueden personalizar con dibujos, patrones matemáticos... Pero también se pueden personalizar otros objetos. Por ejemplo, cortando diseños sobre cinta adhesiva resistente, se pueden pegar estos diseños sobre vasos y aplicar soluciones químicas que realizarán estos grabados en el vidrio.



Figura 6.7: Personalización de objetos [13].

6.4.2.1. Personalización de ropa

Prendas de ropa viejas o dañadas se pueden reciclar jugando con diseños cortados sobre las mismas que dejan ver una segunda capa. Desde camisetas cortas que se superponen a camisetas largas hasta vestidos de boda que tienen cortes con forma de flor, la comunidad maker demuestra que la creatividad es poder, cuidando hasta el último detalle como el traje con chaleco a juego.



Figura 6.8: Personalización de ropa [14].

6.4.3. Diseño y grabado de tableros de juego en madera

Al disponer de la posibilidad de grabar madera, no solo se pueden personalizar objetos, producir cuadros e imágenes artísticas, sino que permite también diseñar creativos juegos de mesa con tableros de diseño único y producirlos con gran calidad. De esta forma es posible disponer de versiones modernas de juegos clásicos ambientados en las historias y personajes favoritos de cada cual.



Figura 6.9: Tableros de juego hechos por grabado laser [18] [19] [20].

6.4.4. Creación de lámparas

Esta herramienta permite el corte de paneles flexibles de plástico, con lo que se pueden crear lámparas de diseño modular. Asimismo, se pueden grabar plásticos más rígidos como el metacrilato y crear lámparas de pie. Como se puede personalizar tanto la forma de la lámpara como los grabados en sus superficies, y tenemos plásticos de múltiples colores, existen ejemplos de grandes diseños de lámparas en la comunidad maker.



Figura 6.10: Lámparas por corte/grabado láser [16] [17].

Capítulo 7

Módulo de herramienta: Ptah

El módulo Ptah, llamado así POR El dios de las artes egipcio, es el módulo de herramienta que utiliza un servomotor para desplazar la herramienta cargada (lápiz, bolígrafo...). Así, esta herramienta permite hacer dibujos con alta precisión. Al final del capítulo se mostrará algún ejemplo de creaciones de la comunidad maker con este tipo de herramientas.

7.1. Hardware específico

El hardware específico que vamos a utilizar va a ser un servomotor y un driver de diseño propio.

7.1.1. Servomotor

Un servomotor es un tipo de motor que consta de un motor de CC, una caja reductora y un circuito de control que permite controlarlo en velocidad y posición. Se controla con una señal PWM y al enviar una señal periódica de 20ms (50 hZ) distintas posiciones están asociadas a distintos anchos de pulso (Ciclos de trabajo).

Los extremos del servomotor del que se dispone están asociados a 0,3ms y 2,3ms, es decir, con 1,3ms posicionaríamos el servomotor en la mitad de su recorrido.

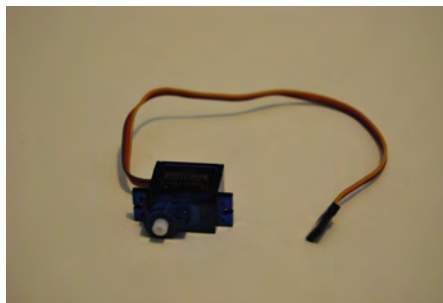


Figura 7.1: Servomotor

7.1.2. Driver

Para la aplicación concreta del servomotor en nuestra herramienta, lo que queremos es que el servomotor se mueva entre dos posiciones para subir y bajar el instrumento de dibujo elegido. Como podemos leer en la bibliografía [12], para esto, y dado que los servomotores se controlan con PWM, podremos controlar nuestro servomotor si generamos una sucesión de pulsos con un timer NE555. Variando el ancho de estos pulsos moveremos el servomotor, y para conseguir esto es suficiente con cambiar una de las resistencias del circuito del timer por un potenciómetro.

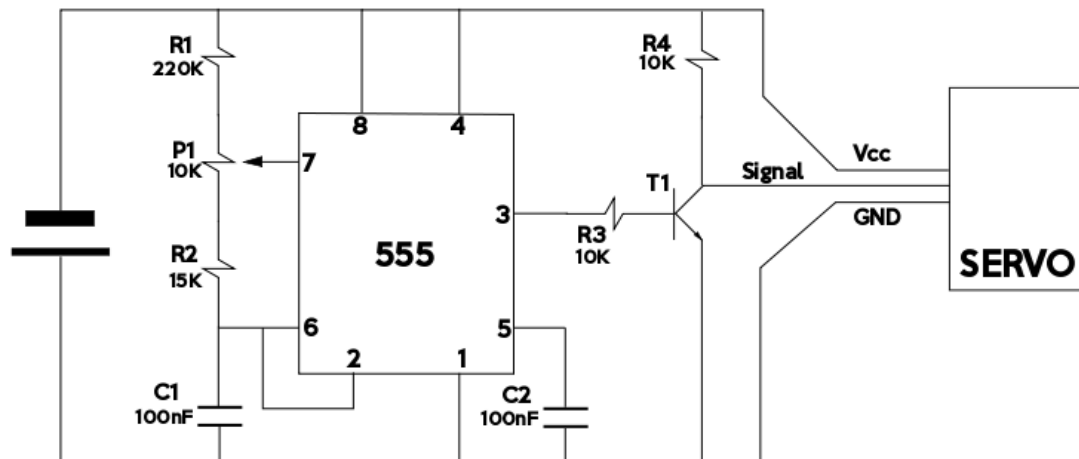


Figura 7.2: Driver con potenciómetro

Como a nosotros nos interesa alternar entre dos posiciones en función de que la señal de control sea 0 o 1, sustituiremos ese potenciómetro por una resistencia del mismo valor y utilizaremos un multiplexor/demultiplexor analógico como el CD4052 conectando sus salidas a los dos terminales de la resistencia. De esta forma lo que conseguimos es que para la entrada $IN = 0$ tenemos $O1 = 1$ y $O2 = 0$, y del mismo modo para $IN = 1$ tenemos $O1 = 0$ y $O2 = 1$. Así la resistencia se hace máxima o 0 en función de la entrada, y variamos entre las dos posiciones del servomotor.

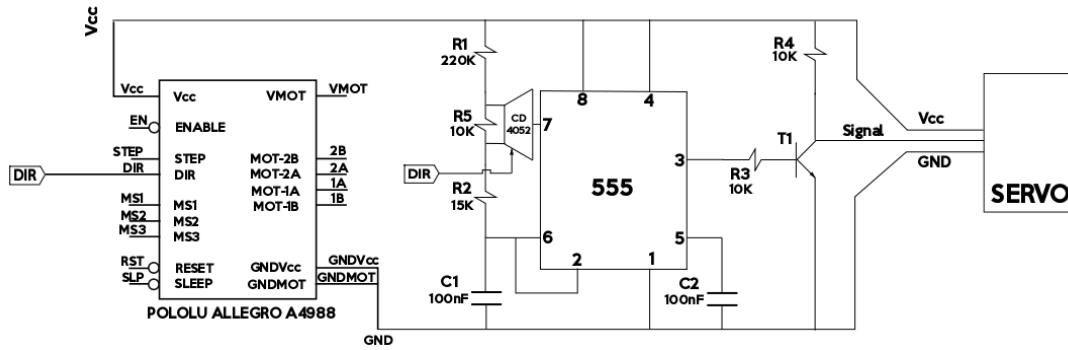


Figura 7.3: Driver con multiplexor/demultiplexor asociado al formato pololu

Como es un prototipo, se ha diseñado el driver del tamaño del driver pololu sobre una placa de prototipado rápido. En futuras versiones del driver se diseñará una PCB específica que ofrezca mayor seguridad, robustez y limpieza.

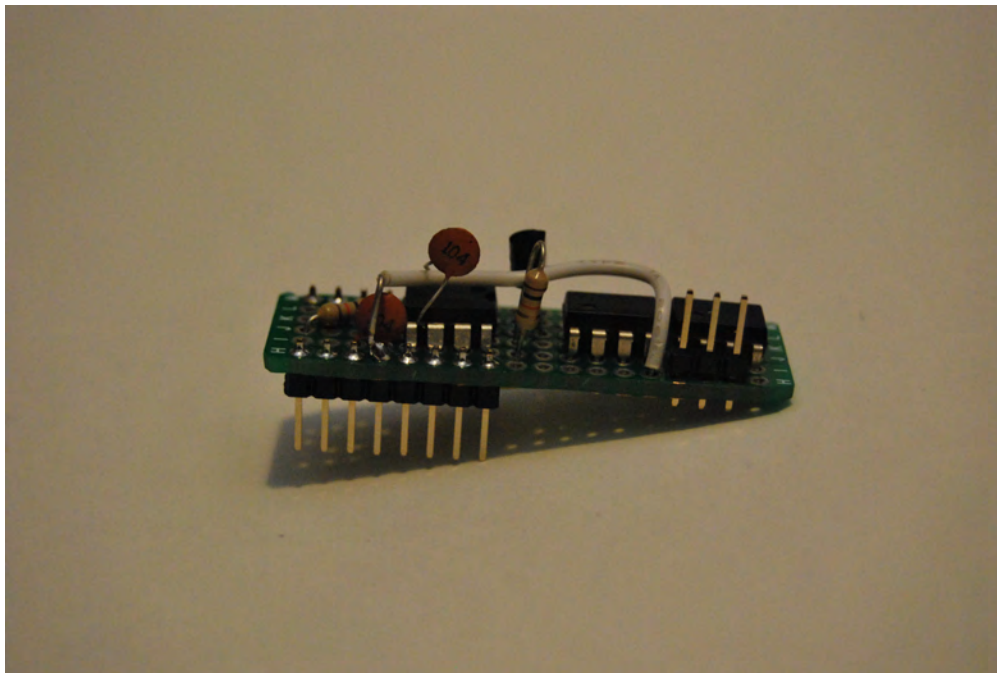


Figura 7.4: Driver para servo montado

7.2. Pieza de agarre al carro de la deslizadera

Esta pieza es la que está atornillada al servomotor y cuando se usa la herramienta Ptah se fija al carro de la deslizadera. De esta forma el servomotor queda fijo y centrado sobre el hueco del carro, por donde pasa la herramienta escogida (lápiz, bolígrafo, rotulador...)

El código de OpenSCAD de esta herramienta se encuentra con el resto de piezas impresas en el apéndice [B](#)

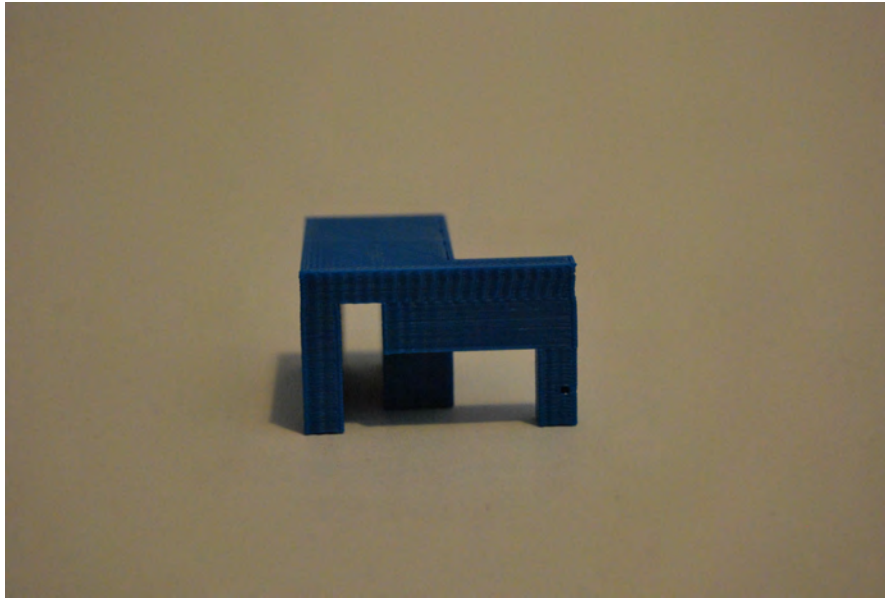


Figura 7.5: Pieza de agarre al carro de Ptah

7.3. Seguridad

Cuando el plóter esté funcionando con esta herramienta, si bien es menos peligroso que el módulo láser, se deben tener en cuenta algunas medidas básicas de seguridad:

- No se debe meter la mano en la máquina mientras esté en funcionamiento para colocar el material de trabajo ni para ningún otro menester. Si se debe hacer cualquier ajuste, se debe parar el plóter, realizar el ajuste y reanudar la marcha del mismo.
- No se debe tocar la electrónica, desconectar o conectar drivers, ajustar los cables de la alimentación, etc...con la fuente de alimentación encendida. Mejor apagarla, realizar las conexiones y comprobaciones pertinentes y posteriormente encenderla.
- Esta herramienta funciona controlando un servomotor, que debe disponer del espacio apropiado para desplazarse de una posición a otra. Colocar barreras

físicas entre estas dos posiciones impidiendo que el servomotor se mueva de la forma en la que está diseñado podría causar daños irreversibles al motor.

7.4. Posibles usos

Esta herramienta tiene una orientación fundamentalmente artística, pues permite realizar mediante CAD dibujos que parecen realizados a mano. Se puede usar con lapiceros, bolígrafos y rotuladores, aunque su uso difiere en el caso de este último. Si bien los lapiceros y los bolígrafos tienen un funcionamiento "todo o nada" son útiles para dibujar planos, o dibujos artísticos e incluso fotografías convertidas a dibujos, con los rotuladores se pueden hacer diseños de puntillismo siempre que se controle el tiempo.



Figura 7.6: Imagen realizada con un Polargraph [21].

Así, un rotulador que se pose en un papel durante un tiempo más corto, generará un punto más pequeño, y uno que se pose más prolongadamente, generará un punto mayor. Mediante software se puede convertir una imagen en un diseño puntillista y realizar este físicamente con esta herramienta y un rotulador.



Figura 7.7: Imagen realizada con un plóter con rotulador [22].

Capítulo 8

Planificación

Es muy importante realizar la planificación de las distintas fases correctamente, pues nos permite ahorrar costes, conocer que fases van a necesitar más trabajo, dirigir eficientemente nuestros esfuerzos y terminar nuestro desarrollo en el plazo acordado. De esta forma, desarrollaremos de forma más competente nuestro proyecto y seremos capaces de satisfacer a nuestros clientes.

Para conseguir este objetivo debemos saber valorar las dimensiones de los problemas a los que nos enfrentamos, la capacidad de los recursos de que disponemos y el tiempo de desarrollo de cada fase, así como el tiempo necesario para cambiar de una tarea a otra o alternar entre tareas. Asimismo es importante conceder un margen de tiempo en nuestra planificación a las tareas críticas, de forma que seamos capaces de completarlas pese a que puedan surgir incidentes de toda índole. Sin embargo, debemos recordar que sobrestimar el tiempo en fases no críticas puede llevarnos a retrasos en la entrega y a ser menos competitivo.

Teniendo esto en cuenta y procurando guardar un equilibrio entre la seguridad de terminar el proyecto a tiempo y la competitividad, es aconsejable dejar un tiempo al final de la planificación para poder cumplir a tiempo con el desarrollo incluso si algunas de las fases se retrasan.

8.1. Planificación inicial

La primera planificación que hemos realizado, de carácter orientativo, dividiendo en fases y tareas el desarrollo de Proteus y asignando tiempos de dedicación específica de nuestros recursos (el personal dedicado al desarrollo de Proteus). Se muestran los tiempos asignados en la siguiente tabla.

Fase	Duración	Recursos
Análisis del problema	48	Desarrollador: Francisco Javier del Álamo
Desarrollo de Proteus	160	Desarrollador: Francisco Javier del Álamo
Desarrollo de la App	40	Desarrollador: Francisco Javier del Álamo
Desarrollo de Surya	48	Desarrollador: Francisco Javier del Álamo
Desarrollo de Pthah	56	Desarrollador: Francisco Javier del Álamo
Memoria	120	Desarrollador: Francisco Javier del Álamo Colaboración Tutor: Andrea Bellucci
Presentación	40	Desarrollador: Francisco Javier del Álamo
Tiempo de seguridad	40	
Total	552	

Estas fases del desarrollo quedan representadas en el siguiente diagrama de Gantt utilizado para estimar la fecha de finalización (incluido en alta resolución en el Anexo 4.1), donde hemos considerado el trabajo de un ingeniero junior trabajando 8 horas diarias. De esta forma hemos determinado que comenzando el 16 de Febrero se terminará el proyecto el 15 de Mayo. Además contaremos con un tiempo de seguridad de 40 horas, con lo que estimaremos la entrega del proyecto en el 22 de Mayo. Como es natural, en el caso de un ingeniero senior los tiempos hubiesen resultado considerablemente inferiores.

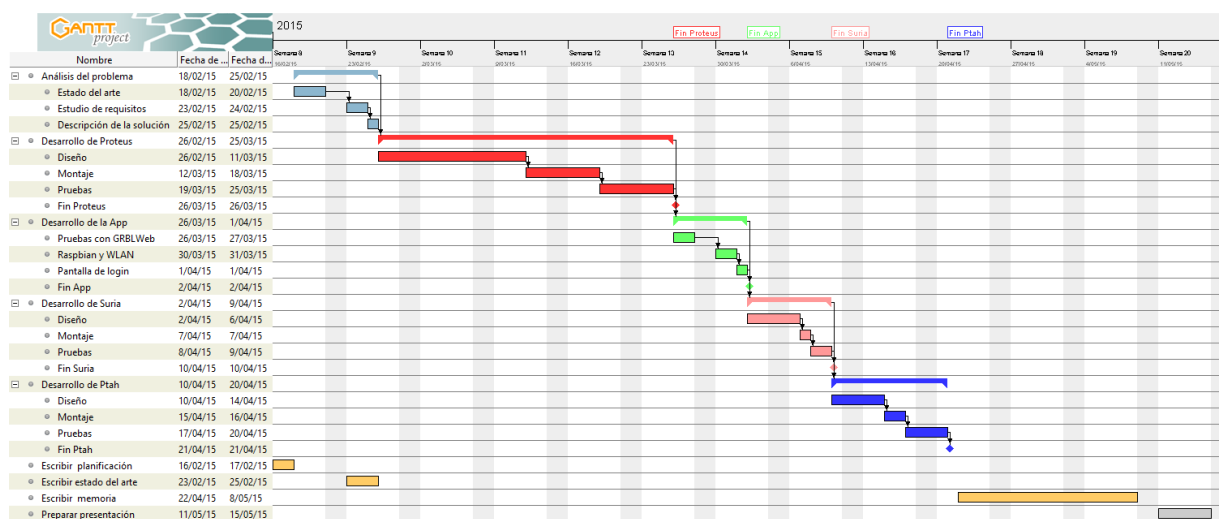


Figura 8.1: Diagrama de Gantt inicial

8.2. Planificación real

La planificación real, donde tenemos en cuenta los datos obtenidos una vez realizadas las distintas fases, difiere ligeramente de la planificación inicial. En la siguiente tabla se muestran los tiempos reales para cada fase.

Fase	Duración	Recursos
Análisis del problema	72	Desarrollador: Francisco Javier del Álamo
Desarrollo de Proteus	176	Desarrollador: Francisco Javier del Álamo
Desarrollo de la App	48	Desarrollador: Francisco Javier del Álamo
Desarrollo de Surya	48	Desarrollador: Francisco Javier del Álamo
Desarrollo de Ptah	64	Desarrollador: Francisco Javier del Álamo
Memoria	136	Desarrollador: Francisco Javier del Álamo Colaboración Tutor: Andrea Bellucci
Presentación	48	Desarrollador: Francisco Javier del Álamo
Total	592	

Si bien en la planificación inicial se contaba con terminar el proyecto el 22 de Mayo, realmente se terminó el 28 de Mayo, lo que implica un retraso de 4 días laborables. Teniendo en cuenta los retos que presentaba inicialmente este proyecto, se ha considerado un retraso aceptable. Se muestra a continuación el correspondiente diagrama de Gantt modificado acorde a los tiempos reales (incluido en alta resolución en el Anexo 4.2).

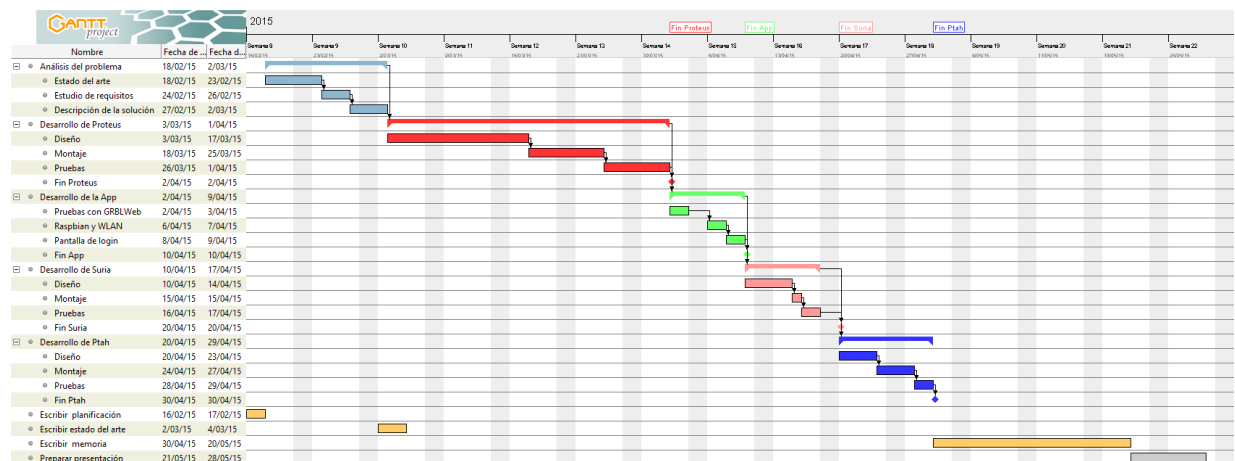


Figura 8.2: Diagrama de Gantt final

Capítulo 9

Presupuesto

9.1. Costes de desarrollo

El coste de desarrollo se compone fundamentalmente de tres partes: el coste de material, el coste de las herramientas y el coste de personal.

Cada una de estas partes se detalla en un apartado, y en estos consta una tabla que muestra los productos, una breve descripción de algunos de ellos, la cantidad, el coste estimado y el coste real, siendo todos los costes expresados en euros. En un último apartado se muestran los costes totales del desarrollo del proyecto, estimados y reales. Finalmente cada apartado se completará con la interpretación del resultado obtenido al comparar el coste estimado con el coste real.

9.1.1. Coste de material

En este apartado se engloban los costes de todos los materiales utilizados para el desarrollo del plóter Proteus.

Se incluyen también los costes del material utilizado para montar el servidor web de la aplicación GRBLWeb.

Además se incluyen los costes del material relativo al desarrollo de las herramientas Surya y Ptah.

Coste estimado de material

Producto	Descripción	Cantidad	Coste unidad (€)	C. total (€)
Arduino UNO	Placa micro-controladora	1	3.87	4
CNC Shield	Interfaz de drivers	1	4.15	4
Driver Pololu	Drivers de motores	3	1.79	5
Sensor final de carrera	Sensor de contacto	3	0.33	1
Motor Nema 17	Motor	3	9	27
Láser	Diodo láser	1	10	10
Gafas de protección	Gafas de protección para el láser	1	5.46	5.46
Raspberry Pi Modelo B versión alpha	Miniordenador	1	20	20
Dongle wifi usb	Usb con wifi	1	1.7	1.7
Tarjeta SD 4GB		1	2.5	2.5
Fuente de alimentación		1	5.5	5.5
Varillas lisas		1	1.8	1.8
Varillas roscadas		2	0.75	1.5
Rodamientos lineales	Rodamiento del carro	1	0.83	0.83
Rodamientos radiales	Rodamientos de los ejes	6	0.41	2.4
Piezas impresas			5	5
Tuercas				3
Arandelas				3
Tornillos				3
Tablón				3
Total				109,69

Tabla 9.1: Coste estimado de material en el proceso de desarrollo

Coste real de material

Producto	Descripción	Cantidad	Coste unidad (€)	C. total (€)
Arduino UNO	Placa micro-controladora	1	3.87	3.87
CNC Shield	Interfaz de drivers	1	4.15	4.15
Driver Pololu	Drivers de motores	3	1.79	5.37
Sensor final de carrera	Sensor de contacto	6	0.41	2.46
Motor Nema 17	Motor	3	8.9	26.7
Láser	Diodo láser	1	10.21	10.21
Soporte del láser con disipador	Disipador de calor para el láser	1	4.18	4.18
Gafas de protección	Gafas de protección para el láser	1	5.46	5.46
Raspberry Pi Modelo B versión alpha	Miniordenador	1	20	20
Dongle wifi usb	Usb con wifi	1	7.8	7.8
Tarjeta SD 4GB		1	2.5	2.5
Fuente de alimentación		1	5.5	5.5
Varillas lisas		1	1.8	1.8
Varillas roscadas		2	0.75	1.5
Rodamientos lineales	Rodamiento del carro	1	0.83	0.83
Rodamientos radiales	Rodamientos de los ejes	6	0.40	2.4
Piezas impresas			10	10
LM317		1	0.25	0.25
Servomotor			2.5	2.5
Componentes de los drivers	Resistencias, condensadores, LM317...		1.5	1.5
Tuercas			7	7
Arandelas			5	5
Tornillos			5	5
Tablón			5	5
Total				140,98

Tabla 9.2: Coste real de material en el proceso de desarrollo

Resultado de la comparación

Los costes reales son algo mayores que los estimados, principalmente debido a tres factores: las piezas que finalmente se compraron de mejor calidad como el USB Wifi, las piezas que se necesitaron durante el desarrollo del proyecto como el servomotor para el módulo de dibujo y el coste algo mayor del estimado de piezas de ferretería.

Aunque se trata de componentes de bajo coste y la diferencia no es muy grande, uno de los objetivos del proyecto es mantener el coste muy bajo, por lo que, si bien la estimación inicial ha resultado útil, pensando en la fabricación de kits se debería tener en cuenta el USB Wifi considerado inicialmente y la compra al por mayor de las piezas de ferretería para acercar más el coste real al coste estimado inicialmente.

9.1.2. Coste de herramientas

A la hora de desarrollar este proyecto era previsible que serían necesarias varias herramientas, las cuales, al poder ser utilizadas posteriormente, han sido tenidas en cuenta en función de su coste de uso.

Coste estimado de herramientas

Producto	Precio (€)	Tiempo de utilización	Coste para el proyecto (€)
Sobremesa HP	450	3 meses	56,25
Makerbot Replicator 2X	2735	1 mes	114
Desarrollador OpenSCAD	18 €/hora	45 horas	810
Total			980,25

Tabla 9.3: Estimación inicial de coste de herramientas

Coste real de herramientas

Producto	Precio (€)	Tiempo de utilización	Coste para el proyecto (€)
Sobremesa HP	450	3 meses	56,25
Makerbot Replicator 2X	2735	1 mes	114
Desarrollador OpenSCAD	18 €/hora	50 horas	900
Total			1070,25

Tabla 9.4: Coste real de herramientas

Resultado de la comparación

El tiempo de diseño de las piezas del plóter fue algo superior a lo estimado debido a las diversas iteraciones realizadas para mejorar las piezas y ajustarlas al comportamiento deseado de las mismas. En cualquier caso, la estimación fue bastante adecuada, pues la desviación es pequeña tanto en coste como en tiempo de desarrollo.

9.1.3. Coste personal

El coste personal utiliza los resultados de las dos planificaciones con una estimación aproximada de los sueldos actuales.

Coste personal estimado

Nombre	Categoría	Salario (€/Hora)	Horas totales	Coste total (€)
Javier del Álamo	Ingeniero Junior	15	552	8280
Total				8280

Tabla 9.5: Coste personal estimado

Coste personal real

Nombre	Categoría	Salario (€/Hora)	Horas totales	Coste total (€)
Javier del Álamo	Ingeniero Junior	15	592	8880
Total				8880

Tabla 9.6: Coste personal real

Resultado de la comparación

Podemos apreciar que en este caso la diferencia es mayor que en los anteriores, debido principalmente a que la planificación inicial no era perfecta y esto afecta al coste de personal.

Una buena planificación nos permite obtener la estimación de coste de personal con mayor precisión, lo cual es particularmente importante debido a que en este tipo de proyectos, estos costes suelen ser los mayores con gran diferencia.

9.1.4. Coste total de desarrollo del proyecto

En la siguiente tabla se compararan los costes totales estimados frente a los reales.

Objeto del Coste	Precio Estimado (€)	Precio Real (€)
Materiales empleados	109,69	140,98
Herramientas utilizadas	980,25	1070,25
Coste Personal	8280	8880
Total	9369.94	10091.23

Tabla 9.7: Costes totales estimados y reales

Resultado de la comparación

Se puede ver que los costes reales superan a los estimados en unos 720 €, lo que no es excesivo considerando el tamaño del proyecto, pero teniendo en cuenta que más del 80 % de este incremento es debido a los costes personales, queda claro que sería conveniente mejorar la estimación de horas de desarrollo en la planificación.

Capítulo 10

Conclusiones

Durante el desarrollo del proyecto he podido usar una impresora 3D, lo que me ha permitido diseñar y fabricar piezas propias para Proteus. Sin esta herramienta se hubiese tenido que optar por un medio de fabricación diferente lo que hubiese incrementado enormemente los costes. Aunque ha sido suficiente para realizar el prototipo, me hubiese venido bien poder utilizar una máquina de creación de PCBS como la Cyclone para los drivers.

Me he dado cuenta de que para realizar un proyecto tan económico capaz de moverse con precisión ha sido necesario aceptar que se desplace a velocidades bajas, por lo que considero que conseguir desplazamientos de la herramienta más veloces sin incrementar excesivamente el coste ni perder precisión es el reto fundamental al que se enfrenta el desarrollo futuro de este tipo de herramientas.

Por otra parte, se espera que la creación de nuevas herramientas haga de esta máquina una herramienta muy versátil. Para crear herramientas compatibles, tan solo hay que crear una pieza como las documentadas en Surya y Ptah, que se puedan encajar en el carro de la deslizadera y estén unidas a la herramienta correspondiente. Algunas herramientas planteadas no necesitan nada más que la herramienta y un driver sencillo, como por ejemplo una herramienta con una cuchilla de vinilo, pero otras, como un tubo de succión para posicionar componentes SMD sobre PCBS podrían requerir de una serie de sensores para identificar donde se encuentra un componente y cogerlo.

Se ha llegado a la conclusión de que es posible construir un plóter de bajo coste si se dispone de las habilidades y conocimientos técnicos de un maker. Si bien es el grupo al que estaba enfocado el proyecto, este también pretende abrir las puertas de este mundo a gente que no disponga de estos conocimientos y habilidades.

Para que estas personas pudiesen construir el plóter se necesita cumplir ciertos requisitos que se planean cubrir en un futuro:

- Deberá distribuirse en forma de kit organizado, con las piezas correctamente identificadas y numeradas.
- En el kit se incluirá un exhaustivo manual de instrucciones con imágenes de cada paso y explicaciones de todas las operaciones que se realicen, sin dar

por supuesto que el cliente tenga conocimientos o habilidades técnicas.

- La electrónica que necesite de soldaduras deberá enviarse soldada, de manera que únicamente se necesite conectar según el manual de instrucciones, y los conectores deberán estar diseñados de tal forma que solo encajen en su posición.
- El conjunto de electrónica Arduino + CNC Shield + Pololus (x3) se enviará montado.
- Los drivers de los motores paso a paso (Pololus) deberán entregarse ya calibrados, listos para ser utilizados tras conectarlos.
- La Raspberry Pi se entregará con el sistema operativo con el servidor modificado ya instalado en la tarjeta de memoria.
- La fuente de alimentación ATX se enviará con las modificaciones realizadas lista para conectarse a la CNC Shield.
- Se adjuntará una plantilla por cada pieza que deba atornillarse a la base para facilitar lo máximo posible el montaje de la estructura del plóter.

10.1. Conclusión personal

Habiendo dedicado a este proyecto una gran cantidad de tiempo y esfuerzo considero que el resultado ha sido, si no plenamente satisfactorio, si en gran medida. Se ha conseguido plantear una solución viable al problema inicial teniendo en cuenta los requisitos especificados y desarrollarla en forma de prototipo.

Considero que este proyecto me ha servido no solo para aprender, me ha hecho aplicar los conocimientos adquiridos durante el grado y entrever la complejidad del desarrollo de un producto y comprender por qué en la industria se trabaja en equipos multidisciplinares.

Desde que comenzó este proyecto este tipo de máquinas de fabricación digital open source ha crecido considerablemente, y es previsible que de aquí a un par de años, esta incipiente revolución esté mucho más integrada en el día a día como una realidad social.

Personalmente estoy contento y satisfecho de haber creado algo, procurando pensar con mimo cada detalle, para otorgar a la comunidad una herramienta que les permita crear y construir, plasmando en la realidad física las increíbles y maravillosas ideas que sus miembros nos brindan.

10.2. Estado actual

A día de hoy el prototipo de Proteus es funcional, junto con la herramienta de grabado láser y la de dibujo.

Además, se ha colaborado al proyecto GRBLWeb implementando un acceso seguro protegido por contraseña. Si los autores aceptan esta funcionalidad estará disponible dentro del propio proyecto.

10.2.1. Requisitos planteados frente a requisitos cumplidos

En la siguiente tabla se muestran de forma comparativa los requisitos planteados originalmente frente a los requisitos que hemos considerado que se han logrado cumplir al final del desarrollo de Proteus.

Tabla 10.1: Tabla de requisitos

Requisito	Cumplido / No cumplido
Muy económico	✓
Open Source	✓
Multiherramienta	✓
Dimensiones flexibles	✓
Portable	✓
Multidispositivo	✓
Fácil uso y construcción	✓
Preciso	✓
Pantalla de login	✓
No requiera ningún conocimiento específico	✗

10.3. Mejoras futuras

Dado que este proyecto aún se encuentra en fase de desarrollo, existen aún varias mejoras que se deben realizar antes de plantearse la comercialización.

10.3.1. Crear un diseño portable para grandes medidas

Profundizando en la alternativa de diseño propuesta en el punto 3.2.4. considero interesante desarrollar una estructura modular y portable para un plóter de grandes dimensiones. Esto implica el rediseño de los soportes de la deslizadera, pues esta deberá ser también modular. Posiblemente lo más apropiado sea que los soportes se rediseñen para convertirse en piezas de agarre y que la deslizadera completa se pueda poner y quitar apretando unos pocos tornillos o simplemente encajándola.

Además, sería conveniente crear una superficie para el área de trabajo que pudiese acoplarse y desacoplarse con sencillez, de manera que se permita el desarrollo de bases compatibles, como por ejemplo una base rotativa para grabar elementos cilíndricos.

10.3.2. Diseñar los drivers de las herramientas como PCBS listas para soldar

Diseñando las PCBS de los drivers se conseguiría reducir el espacio, además de reducir costes y hacer el diseño más robusto. Este paso es, como se ha comentado anteriormente, fundamental para popularizar el uso del plóter. Además, al conservar el formato del driver pololu, estos drivers son compatibles con la mayoría de opciones de electrónica de las impresoras 3D open source, por lo que las herramientas así diseñadas pueden llevarse fácilmente a este otro tipo de máquinas de fabricación digital doméstica.

10.3.3. Crear una red inalámbrica local que lleve directamente a la aplicación web

Creando una WLAN llamada "Proteus Plotter".^{en} la Raspberry Pi y creando un portal cautivo mediante la redirección de los DNS a la webapp utilizando bind9 y modificando el archivo de configuración. También existe la posibilidad de utilizar uno de los muchos portales cautivos disponibles para Linux y modificarlo.

Así, una vez conectados a la red "Proteus Plotter" Si nos dirigimos desde un navegador web a www.proteusplotter.com el servidor nos dirigirá automáticamente a la pantalla de login de la aplicación que controla el plóter.

Bibliografía

- [1] Consultada 26 de Febrero 2015, en <http://www.lasersaur.com>.
- [2] Consultada 26 de Febrero de 2015, en <http://www.piccolo.cc/#Build-Your-Own>.
- [3] Consultada 26 de Febrero de 2015, en <http://www.shapeoko.com>.
- [4] Consultada 26 de Febrero de 2015, en <http://www.instructables.com>.
- [5] Consultada de 26 de Febrero 2015, en <https://github.com/carlogsg/Cyclone-PCB-Factory>.
- [6] Consultada 1 de Marzo de 2015, en <http://www.arduino.cc/en/Main/ArduinoBoardUno>.
- [7] Consultada 1 de Marzo de 2015, en <http://www.atmel.com/Images/doc8161.pdf>.
- [8] Consultada 1 de Marzo de 2015, en <http://blog.protoneer.co.nz/arduino-cnc-shield/>.
- [9] Consultada 1 de Marzo de 2015, en <https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md>.
- [10] Consultada 2 de Marzo de 2015, en <https://www.pololu.com/product/2980>.
- [11] Consultada 2 de Marzo de 2015, en <http://www.electronicaembajadores.com/Datos/pdf1/sm/smci/a4988.pdf>.
- [12] Consultada 20 de Abril de 2015, en <http://mimaquetaz.blogspot.com.es/2011/11/llegan-los-servos-i.html>.
- [13] Consultada 17 de Abril de 2015, en <http://www.instructables.com/id/How-to-make-laser-cut-jewelry-for-Valentines-Day/?ALLSTEPS>.
- [14] Consultada 17 de Abril de 2015, en <http://www.instructables.com/id/Laser-Cut-Wedding-Dress/>.
- [15] Consultada 17 de Abril de 2015, en <http://www.instructables.com/id/Custom-PCB-Prototyping-using-a-Laser-Cutter/>.

- [16] Consultada 17 de Abril de 2015, en <http://www.instructables.com/id/Laser-Cutting/?ALLSTEPS>.
- [17] Consultada 17 de Abril de 2015, en <http://www.instructables.com/id/How-to-make-a-laser-cut-lamp/>.
- [18] Consultada 17 de Abril de 2015, en <http://www.instructables.com/id/Laser-cut-Marquetry/>.
- [19] Consultada 17 de Abril de 2015, en <http://www.instructables.com/id/Creating-Custom-Cribbage-Boards/>.
- [20] Consultada 17 de Abril de 2015, en <http://www.instructables.com/id/Custom-Laser-Cut-Catan-Board/>.
- [21] Consultada 29 de Abril de 2015, en <http://www.instructables.com/id/Polarograph-Drawing-Machine/>.
- [22] Consultada 29 de Abril de 2015, en <http://www.instructables.com/id/Dot-Matrix-Printer-from-a-CDDVD-Reader-with-Arduin/>.

Apéndice A

Anexo1: Servidor de GRBLWeb en la Raspberry Pi con login

Software del servidor en la Raspberry Pi: Aplicación web controladora del plóter.

Para controlar los movimientos del plóter desde cualquier dispositivo se ha optado por integrar la solución GRBLWeb de XYZprinters. Este software lanza la aplicación web sobre un servidor de node.js a la que podemos acceder en la ip asignada por el router a la Raspberry Pi.

Alternativamente al router podemos utilizar un dispositivo USB que de capacidad wifi a la Raspberry Pi, de forma que todo el servidor web de la aplicación estará contenido en el mismo dispositivo, necesitando únicamente una fuente de alimentación.

Su instalación es sencilla, solo debemos dirigirnos a xyzbots.com/grblweb.html, descargar el tipo de archivo correspondiente (En nuestro caso el zip) y flashearlos a la tarjeta SD de la Raspberry Pi. Al conectarlo a internet se le otorgará una ip, y accediendo a esta podremos acceder a la aplicación web.

Para nuestros propósitos vamos a necesitar ir un poco más allá, concretamente tendremos que acceder al sistema operativo que usa GRBLWeb, Raspbian. Para ello deberemos loguearnos con `user:pi` y `password:raspbian`.

A continuación ejecutamos `passwd` y cambiamos la contraseña a otra de nuestra elección, en este caso se ha elegido "proteus plotter". Después ejecutamos `startx` y se iniciará el modo gráfico. En la carpeta de usuario `/home/pi` encontraremos la carpeta `grblweb` que cuenta con el servidor de node.js de la aplicación (`server.js`).

Modificaremos ligeramente el servidor haciendo que exista una pantalla de login que de paso a la aplicación. Esta pantalla de login será un servidor de node.js dentro del original, que llamará a un html y css que contienen los formularios para la identificación de usuario y en caso de confirmar un acceso correcto llamará al servidor de la webapp original.

El código html que se ha guardado en un archivo llamado `index.html` es:

```
<!DOCTYPE html>

<html>

  <head>

    <title>Proteus Plotter</title>

    <link rel="stylesheet" href="css/style.css" media="screen"
    title="no title" charset="utf-8">

  </head>

  <body>

    <div id="title_wrapper">

      <div id="proteus"></div>

      <h1>Proteus Plotter</h1>

    </div>

    <form id="login" method="post" action="/">

      <input id="user_input" type="text" name="user" placeholder="User">

      <input id="password_input" type="password" name="password"
      placeholder="Password">

      <input id="button" type="submit" value="submit">

    </form>

  </body>

</html>
```

La hoja de estilos para el html anterior se ha guardado en un archivo llamado style.scss que genera el código css. Este es:

```
//out: ../css/style.css

// fonts
@import url(http://fonts.googleapis.com/css?family=Raleway:400,600,300);
@import url(http://fonts.googleapis.com/css?family=Montserrat);
@import url(http://fonts.googleapis.com/css?family=Oswald:400,300);

// variables
$black: #444;
$white: #f5f5f5;
$primary_font: "Oswald";
$secondary_font: "Montserrat";
$input_font: "Raleway";

* {
  margin: 0;
  padding: 0;
}

body {
  background: $white;
  text-align: center;

  #title_wrapper {
    display: block;
    margin: 0 auto;
    margin-top: 100px;
    text-align: center;

    #proteus {
      display: inline-block;
      border-radius: 50%;
      width: 120px;
      height: 120px;
      vertical-align: middle;
      background-image: url(../img/proteus.gif);
      background-size: 130px 130px;
      background-position: center center;
      box-shadow: 0 2px 5px 0 rgba(0, 0, 0, 0.26);
    }

    h1 {
      display: inline-block;
      font-family: $primary_font;
    }
  }
}
```



```
        color: $black;
        font-weight: 300;
        text-transform: uppercase;
        vertical-align: middle;
        margin-left: 20px;
        letter-spacing: 2px;
    }
}

form {
    display: block;
    margin: 80px auto 0 auto;
    width: 90%;
    max-width: 500px;

    input {
        display: block;
        margin: 20px auto 0 auto;
        padding: 7px 12px;
        font-family: $input_font;
        font-size: 16px;
        border: 1px solid rgba(0, 0, 0, 0.2);
    }

    #button {
        padding: 5px 10px;
        background: $white;
        border: 2px solid $black;
        border-radius: 30px;
        cursor: pointer;
        font-family: $secondary_font;
        text-transform: uppercase;
        font-size: 12px;
        color: $black;
        margin-top: 30px;

        transition: .3s all ease-in-out;

        &:hover {
            background: $black;
            color: $white;
        }
    }
}
```

Luego se ha modificado el archivo server.js incluyendo un servidor para el login al inicio:

```
//  
// LOGIN  
//  
var express = require('express'),  
    server = express(),  
    path = require('path'),  
    bodyParser = require('body-parser');  
  
// static files  
server.use(express.static(path.join(__dirname + '/login/')));  
  
// body parser for getting the form information  
server.use(bodyParser.json());  
server.use(bodyParser.urlencoded({  
    extended: true  
}));  
  
// show login index  
server.get('/', function(request, response) {  
    response.sendFile(path.join(__dirname + '/login/index.html'));  
});  
  
// when submitting form  
server.post('/', function(request, response) {  
    console.log(request.body);  
  
    if (request.body.user == 'admin' && request.body.password == '1234') {  
        start_grbl();  
        setTimeout(function () {  
            console.log();  
        }, 2000);  
        response.redirect('http://localhost:8000');  
    }  
});  
  
// listen on 8080  
server.listen(8080, function() {  
    console.log('listening on 8080');  
});
```

y se ha puesto el código de la aplicación web dentro de una función que es llamada solo si el usuario y la contraseña introducidos son correctos:

```
function start_grbl() {  
}
```

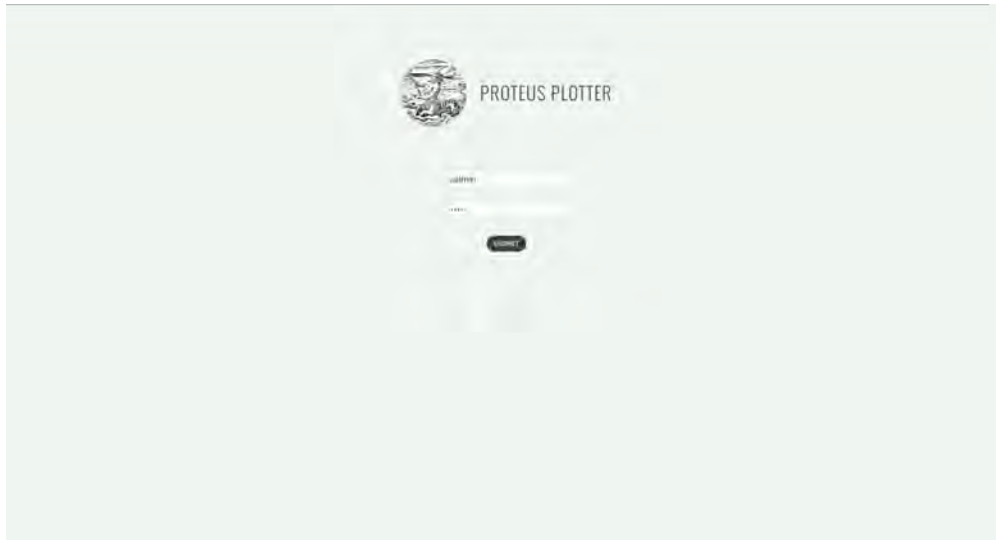


Figura A.1: Pantalla de login añadida a GRBLWeb

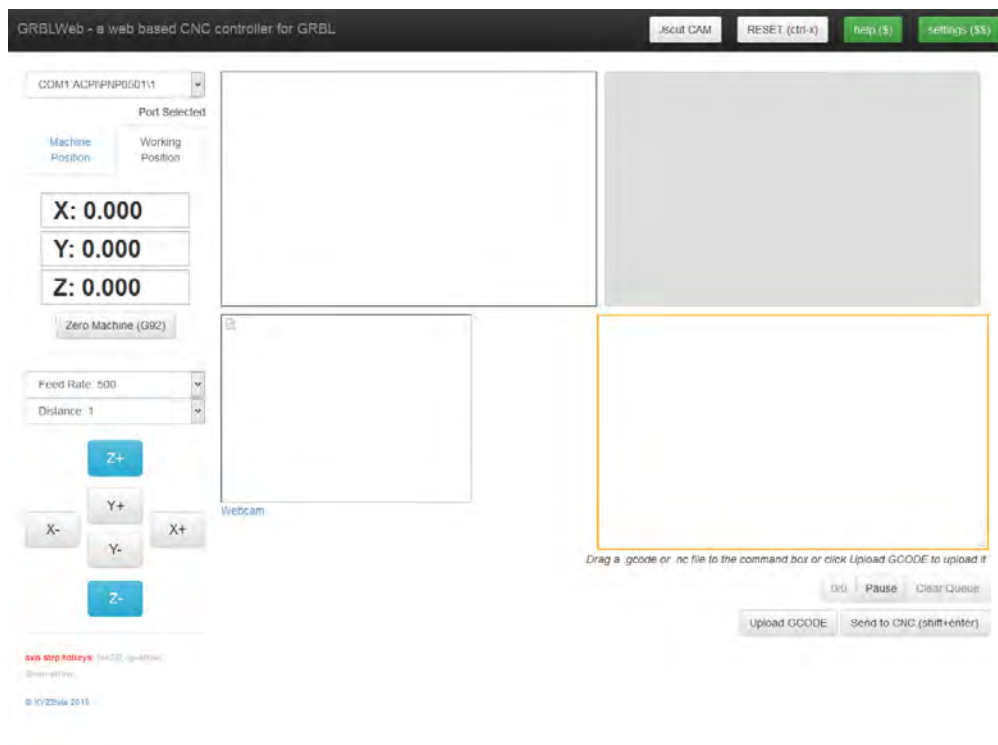


Figura A.2: Pantalla de la aplicación GRBLWeb

Apéndice B

Anexo2: Piezas impresas diseñadas en Open SCAD

B.1. Piezas de Proteus

B.1.1. Cojinetes para los rodamientos: Rod.scad

```
difference(){  
  
// Body  
    union(){  
        cylinder(r=15,h=12,$fn=50);  
  
        translate([-15,0,0])  
        cube([15*2,21,12]);  
  
        translate([-15,0,12])  
        cube([5,21,12]);  
  
        translate([10,0,12])  
        cube([5,21,12]);  
  
        translate([-24,16,0])  
        cube([48,5,12]);  
  
        translate([-24,21,6])  
        rotate([90,0,0])  
        scale([0.7,1,1])  
        cylinder(r=6,h=5,$fn=50);  
  
        translate([24,21,6])  
        rotate([90,0,0])  
        scale([0.7,1,1])
```

```
        cylinder(r=6,h=5,$fn=50);
    }

// Rod
    translate([0,0,3])
    cylinder(r=11.15,h=15);

// Rod hole
    translate([0,0,-20])
    cylinder(r=10,h=40);

// Hole 1
    translate([-22.2,23,6])
    rotate([90,0,0])
    cylinder(r=2,h=12,$fn=10);

// Hole 2
    translate([22.2,23,6])
    rotate([90,0,0])
    cylinder(r=2,h=12,$fn=10);

// Support
    translate([-17,0,12])
    rotate([30,0,0])
    cube([34,25,12]);

}
```

B.1.2. Soporte superior de los motores: top.scad

```
    rotate([180,0,0])
    translate([0,0,-65])
    difference(){

        translate([0,0,50])
        color([1,0,0])

// Body
        union(){
            translate([-24,-21.1,0])
            cube([48,45,12]);

            translate([-35,-21.1,0])
            cube([70,5,12]);
        }
    }
```

```
// Holes
union(){
    translate([-21.1,-22.1,0])
    cube([42.2,43.2,60]);

    translate([0,0,40])
    cylinder(r=11,h=30);

    translate([15.5,15.5,40])
    cylinder(r=1.5,h=30,$fn=10);

    translate([15.5,-15.5,40])
    cylinder(r=1.5,h=30,$fn=10);

    translate([-15.5,15.5,40])
    cylinder(r=1.5,h=30,$fn=10);

    translate([-15.5,-15.5,40])
    cylinder(r=1.5,h=30,$fn=10);

    translate([28,16,48])
    rotate([0,0,45])
    cube(20);

    translate([-28,16,48])
    rotate([0,0,45])
    cube(20);

    translate([28,-8,56])
    rotate([90,0,0])
    cylinder(r=1.5,h=30,$fn=10);

    translate([-28,-8,56])
    rotate([90,0,0])
    cylinder(r=1.5,h=30,$fn=10);
}

}
```

B.1.3. Soporte inferior del motor derecho: bottomright.scad

```
rotate([180,0,0])
translate([0,0,-65])

difference(){
```

```
        translate([0,0,50])
        color([1,0,0])
// Body
    union(){
        translate([-24,-21.1,0])
        cube([48,45,12]);

        translate([-35,-21.1,0])
        cube([70,5,12]);
    }
// Holes
    union(){
        translate([-21.1,-22.1,0])
        cube([42.2,43.2,60]);

        translate([28,16,48])
        rotate([0,0,45])
        cube(20);

        translate([-28,16,48])
        rotate([0,0,45])
        cube(20);

        translate([28,-8,56])
        rotate([90,0,0])
        cylinder(r=1.5,h=30,$fn=10);

        translate([-28,-8,56])
        rotate([90,0,0])
        cylinder(r=1.5,h=30,$fn=10);

        translate([-23,0,54])
        cube(10,center=true);
    }
}
```

B.1.4. Soporte inferior del motor izquierdo: bottom-left.scad

```
rotate([180,0,0])
translate([0,0,-65])
difference(){
```

```
        translate([0,0,50])
        color([1,0,0])
// Body
    union(){
        translate([-24,-21.1,0])
        cube([48,45,12]);

        translate([-35,-21.1,0])
        cube([70,5,12]);
    }

// Holes
    union(){
        translate([-21.1,-22.1,0])
        cube([42.2,43.2,60]);

        translate([28,16,48])
        rotate([0,0,45])
        cube(20);

        translate([-28,16,48])
        rotate([0,0,45])
        cube(20);

        translate([28,-8,56])
        rotate([90,0,0])
        cylinder(r=1.5,h=30,$fn=10);

        translate([-28,-8,56])
        rotate([90,0,0])
        cylinder(r=1.5,h=30,$fn=10);

        translate([23,0,54])
        cube(10,center=true);
    }
}
```

B.1.5. Unión motor-varilla superior: uniontop.scad

```
difference(){
// Body
    translate([0,0,1])
    cube([22,26,16],center=true);
```



```
// Holes
union(){

    translate([-1,0,0])
    rotate([0,90,0])
    cylinder(r=3,h=20,$fn=25);
    rotate([-90,0,90])
    cylinder(r=4.5,h=30,$fn=25);

    translate([7,8,0])
    cylinder(r=2,h=20,center=true);

    translate([7,-8,0])
    cylinder(r=2,h=20,center=true);

    translate([-7,8,0])
    cylinder(r=2,h=20,center=true);

    translate([-7,-8,0])
    cylinder(r=2,h=20,center=true);

    translate([0,0,20])
    cube([40,40,40],center=true);
}

}
```

B.1.6. Unión motor-varilla inferior: unionbottom.scad

```
difference(){
// Body
    translate([0,0,1])
    cube([22,26,16],center=true);

// Holes
    union(){

        translate([-1,0,0])
        rotate([0,90,0])
        cylinder(r=3,h=20,$fn=25);

        rotate([-90,0,90])
        cylinder(r=4.5,h=30,$fn=25);

        translate([7,8,0])
```

```
cylinder(r=2,h=20,center=true);

translate([7,-8,0])
cylinder(r=2,h=20,center=true);

translate([-7,8,0])
cylinder(r=2,h=20,center=true);

translate([-7,-8,0])
cylinder(r=2,h=20,center=true);

translate([5,0,0])
cylinder(r=2,h=20,center=true);

translate([5,0,2])
cylinder(r=3,h=8,center=true,$fn=6);

translate([0,0,-20])
cube([40,40,40],center=true);
}

}
```

B.1.7. Soporte izquierdo de la deslizadera: leftsupport.scad

```
// Body

difference(){

    rotate([0,-90,0])
    difference(){

        translate([0,0,8])
        cube([22,24,80],center=true);

        union(){
            scale(1.05)
            import("Hex_nut_M8.stl",convexity=10);

            cylinder(r=4.2, h=150,center=true);

            translate([0,0,3.25])
            cylinder(r=4.35, h=7.2,center=true);

            translate([0,-6.65,0])
```

```
        cube([13.3,13.3,6.8]);

        translate([31,0,35])
        rotate([0,90,0])
        cylinder(r=4.2, h=50,center=true);

    }

}

    translate([-18,-15,-12])
    color([0,0,1])
    cube([55,5,25]);

}

// Top end

    translate([27,7,-11])
    cube([5,15,5]);

// Bottom end

    translate([-48,7,-11])
    cube([5,15,5]);
```

B.1.8. Soporte derecho de la deslizadera: rightsupport.scad

```
// Body

    rotate([0,-90,0])
    difference(){

        translate([0,0,8])
        cube([22,24,80],center=true);

        union(){
            scale(1.05)
            import("Hex_nut_M8.stl",convexity=10);

            cylinder(r=4.2, h=150,center=true);

            translate([0,0,3.25])
            cylinder(r=4.35, h=7.2,center=true);
```

```
        translate([0,-6.65,0])
        cube([13.3,13.3,6.8]);

        translate([31,0,35])
        rotate([0,90,0])
        cylinder(r=4.2, h=50,center=true);

    }

}

// Top end

    translate([27,-22,-11])
    cube([5,15,5]);

// Bottom end

    translate([-48,-22,-11])
    cube([5,15,5]);
```

B.1.9. Deslizadera de la herramienta: toolsupport.stl

```
difference(){
    union(){

//Rod
        translate([40,0,0])
        rotate([90,0,0])
        cylinder(r=10,h=30,$fn=20,center=true);

//Body
        translate([-10,-7,-10])
        cube([20,14,55]);

        translate([20,0,0])
        cube([40,30,20],center=true);

        translate([40,0,0])
        rotate([90,0,0])
        cylinder(r=10,h=30,$fn=20,center=true);

        translate([47,0,15])
        cube([6,30,30],center=true);
```

```
    }

    union(){

// HexNut
        translate([0,0,35.25])
        rotate([0,270,90])
        translate([0,0,-3.4])
        union(){
            scale(1.05)
            import("Hex_nut_M8.stl",convexity=10);

            translate([0,0,3.25])
            cylinder(r=4.35, h=7.2,center=true);

            translate([0,-6.65,0])
            cube([13.3,13.3,6.8]);
        }

//Up bar
        translate([0,0,35.25])
        rotate([90,0,0])
        cylinder(r=4,h=100,$fn=20,center=true);

//Rod
        translate([40,0,0])
        rotate([90,0,0])
        cylinder(r=7.5,h=50,$fn=20,center=true);

//Hole

        translate([20,0,0])
        cube([20,20,40],center=true);

    }
}
```

B.1.10. Pieza de sujección del láser: laserholder.stl

```
    difference(){
// Body
        cube([23,34,25]);

        translate([0,2,0])
        union(){
```

```
// Grip
    translate([11,-3,-5])
    cube([6,36,25]);

    translate([-10,7,-5])
    cube([26,16,25]);

// Screwholes

    translate([5,3,10])
    rotate([0,90,0])
    cylinder(r=2,h=20,$fn=20,center=true);

    translate([5,27,10])
    rotate([0,90,0])
    cylinder(r=2,h=20,$fn=20,center=true);
}
}
```

B.1.11. Pieza de sujeción del servo: servoholder.stl

```
difference(){
// Body
    union(){
        cube([23,34,25]);

        translate([-18,29,0])
        cube([35,5,25]);
    }
    translate([0,2,0])
    union(){
// Grip
        translate([11,-3,-5])
        cube([6,36,25]);

// Holes
        translate([-12,22,-1])
        cube([24,17,13]);

        translate([-16.2,30,5.5])
        rotate([90,0,0])
        cylinder (r=0.75,h=15,$fn=15,center=true);
    }
}
```

Apéndice C

Anexo3: Diagrama de la electrónica

Dado que la imagen del diagrama de la electrónica del Capítulo 5 tiene unas dimensiones muy reducidas, en este anexo se presenta ampliado para mayor legibilidad.

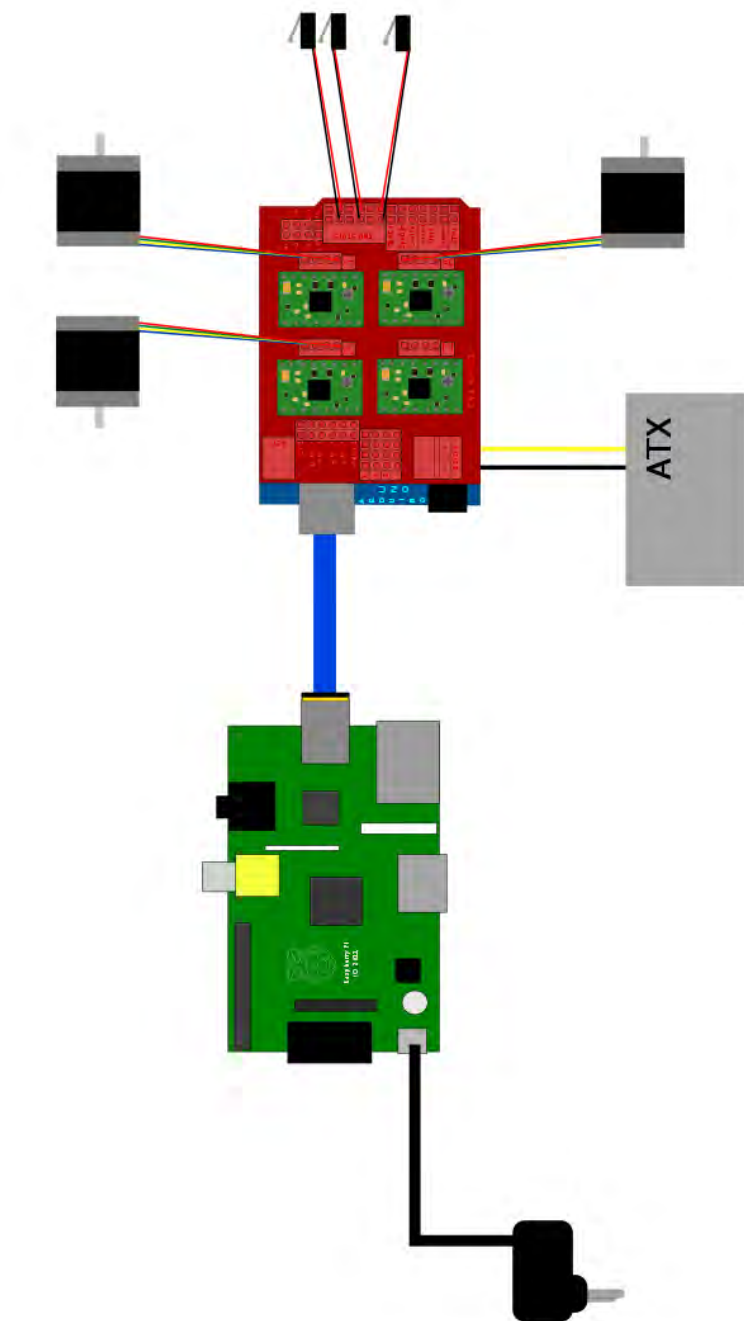


Figura C.1: Diagrama de la electrónica ampliado

Apéndice D

Anexo4: Diagramas de Gantt

Por mayor claridad expositiva, se presentan ampliados a continuación y de forma apaisada los diagramas de Gantt de la planificación (Capítulo 8)

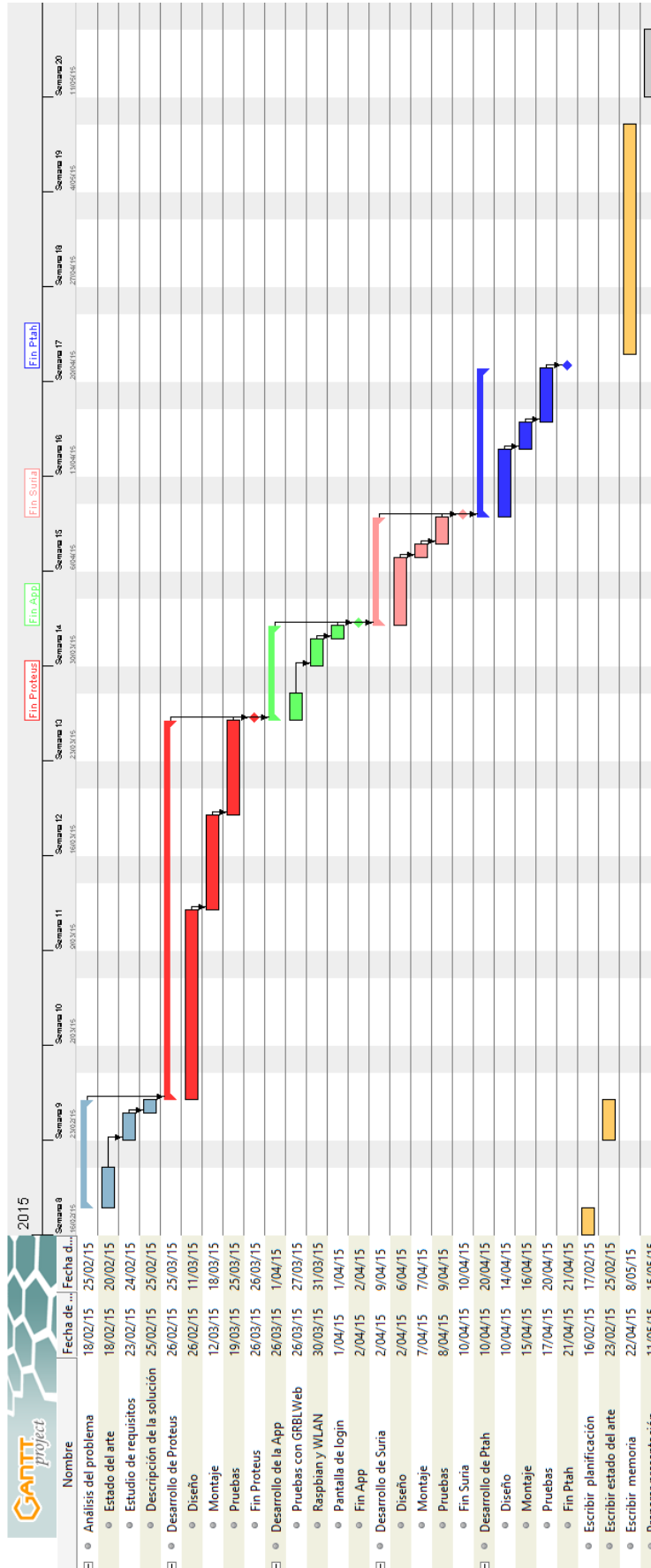


Figura D.1: Diagrama de Gantt inicial ampliado

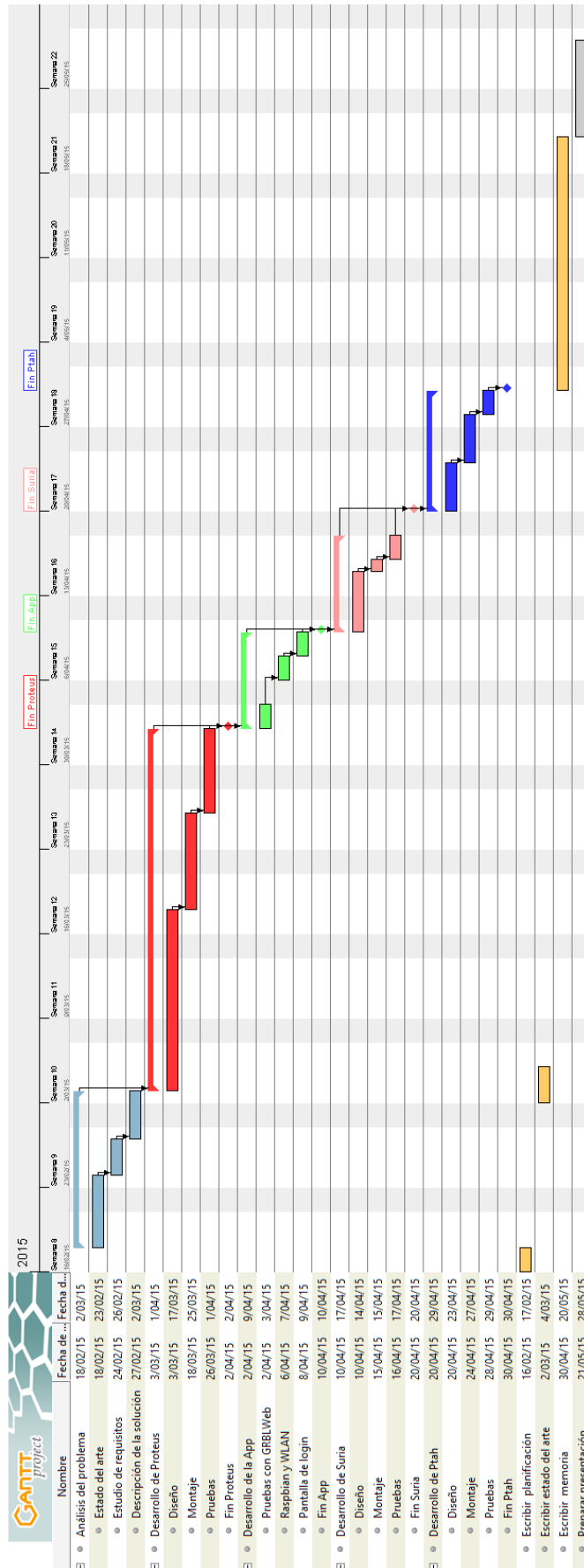


Figura D.2: Diagrama de Gantt final ampliado

Índice de figuras

3.1. Ploter polargraph	18
3.2. Ploter pantografo	19
3.3. Ploter paralex	19
3.4. Ploter X-Y fijo	20
3.5. Ploter X-Y portable	21
4.1. Placa microcontroladora Arduino UNO.	26
4.2. CNC Shield.	27
4.3. Driver A4988 de motor paso a paso.	28
4.4. Motor paso a paso NEMA17.	29
4.5. Fuente de alimentación	30
4.6. Miniordenador Raspberry Pi B modelo de 256Mb.	31
4.7. Adaptador USB Wifi	32
4.8. Transformador de la Raspberry Pi.	32
5.1. Cojinetes impresos para los rodamientos radiales	34
5.2. Anclaje superior de los motores.	34
5.3. Anclaje inferior del motor derecho.	35
5.4. Anclaje inferior del motor izquierdo	35
5.5. Parte superior de la unión del motor con la varilla roscada	36
5.6. Parte inferior de la unión del motor con la varilla roscada	36
5.7. Parte izquierda de la deslizadera.	37
5.8. Parte derecha de la deslizadera.	37
5.9. Carro de la deslizadera	38
5.10. Cambio de diseño en los cojinetes	39
5.11. Potenciómetro y V_{ref} para calibrar la intensidad que circula por el driver	41

5.12. Esquema con las conexiones de toda la electrónica	41
6.1. Láser	43
6.2. Driver para láser controlado	43
6.3. Driver para láser controlado asociado al formato pololu	44
6.4. Driver para láser montado	44
6.5. Pieza de agarre al carro de Surya	45
6.6. Creación de PCBS [15].	46
6.7. Personalización de objetos [13].	47
6.8. Personalización de ropa [14].	47
6.9. Tableros de juego hechos por grabado laser [18] [19] [20].	48
6.10. Lámparas por corte/grabado láser [16] [17].	48
7.1. Servomotor	49
7.2. Driver con potenciómetro	50
7.3. Driver con multiplexor/demultiplexor asociado al formato pololu . .	51
7.4. Driver para servo montado	51
7.5. Pieza de agarre al carro de Ptah	52
7.6. Imagen realizada con un Polargraph [21].	53
7.7. Imagen realizada con un plóter con rotulador [22].	54
8.1. Diagrama de Gantt inicial	56
8.2. Diagrama de Gantt final	57
A.1. Pantalla de login añadida a GRBLWeb	74
A.2. Pantalla de la aplicación GRBLWeb	74
C.1. Diagrama de la electrónica ampliado	87
D.1. Diagrama de Gantt inicial ampliado	89
D.2. Diagrama de Gantt final ampliado	90

Índice de tablas

2.1. Máquinas similares a Proteus A	14
2.2. Máquinas similares a Proteus B	14
2.3. Máquinas similares a Proteus C	14
2.4. Máquinas similares a Proteus D	14
4.1. Conexiones de Fuente ATX	30
9.1. Coste estimado de material en el proceso de desarrollo	59
9.2. Coste real de material en el proceso de desarrollo	59
9.3. Estimación inicial de coste de herramientas	60
9.4. Coste real de herramientas	60
9.5. Coste personal estimado	61
9.6. Coste personal real	61
9.7. Costes totales estimados y reales	62
10.1. Tabla de requisitos	65